

NASA Grant Number NGR 34-002-193

SURFACE FITTING THREE-DIMENSIONAL BODIES

by

Fred R. DeJarnette

Mechanical and Aerospace Engineering Dept.

North Carolina State University

Raleigh, N. C. 27607

for

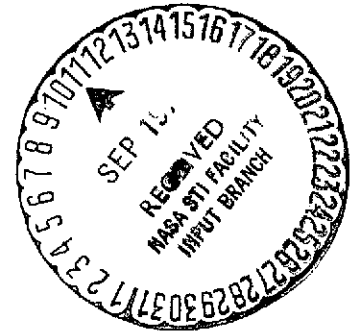
Langley Research Center

National Aeronautics and Space Administration

Hampton, Virginia 23665

Final Report, covering the period May, 1973 - July, 1974

(NASA-CR-139663) SURFACE FITTING THREE-DIMENSIONAL BODIES Final Report, May 1973 - Jul. 1974 (North Carolina State Univ.) 82 p HC \$7.25 CSDL 12A	N74-32039  Unclas G3/19 47742
--	--



## FOREWORD

This research was supported by NASA Langley Grant No. NGR 34-002-193, and the NASA Technical Officer for this grant was Mr. H. Harris Hamilton, Thermal Analysis Section, NASA Langley Research Center.

## ACKNOWLEDGMENTS

The author would like to express his appreciation to the following people for their assistance:

1. Mr. H. Harris Hamilton for his many helpful suggestions and review of the manuscript.
2. Mr. Charles P. Ford, graduate student at North Carolina State University, for his assistance in the curve-fitting and programming.
3. Ms. Marjorie Paulsen for typing and printing this report.

# CONTENTS

	Page
FOREWORD . . . . .	iii
ACKNOWLEDGMENTS . . . . .	iii
SUMMARY . . . . .	1
INTRODUCTION . . . . .	2
SYMBOLS . . . . .	5
ANALYSIS . . . . .	7
Curve-Fit in a Cross-Sectional Plane . . . . .	7
Longitudinal Variation of Cross Sections . . . . .	13
Geometry in Polar Coordinates . . . . .	15
COMPUTATIONAL ALGORITHM . . . . .	16
APPLICATION TO 70° DELTA WING . . . . .	18
CONCLUDING REMARKS . . . . .	19
Appendix A Solution for Conic Section Equations . . . . .	21
Appendix B Least-Squares Solution for Conic Section Coefficients . . . . .	24
Appendix C Description of Program . . . . .	27
Main Program for Determining Geometry . . . . .	27
Main Program for Using Geometry Subroutine Package . . . . .	28
Subroutine GEOM . . . . .	28
Subroutine INGeo . . . . .	29
Subroutine LSTSQ . . . . .	29
Function FZ . . . . .	30
Subroutine SPLX . . . . .	30
Subroutine SPLINE . . . . .	31
Subroutine FCN . . . . .	32
Subroutine DERIV . . . . .	33

Subroutine LINEQ . . . . .	33
Subroutine SIMQ . . . . .	33
Description of Input . . . . .	35
Description of Output . . . . .	41
Program Listing . . . . .	43
Appendix D Input and Output for 70° Delta Wing . . . . .	60
REFERENCES . . . . .	72
FIGURES . . . . .	73

NASA Grant Number NGR 34-002-193

## SURFACE FITTING THREE-DIMENSIONAL BODIES

by Fred R. DeJarnette

North Carolina State University

### SUMMARY

The geometry of general three-dimensional bodies is generated from coordinates of points in several cross sections. Since these points may not be smooth, they are divided into segments and general conic sections are curve fit in a least-squares sense to each segment of a cross section. The conic sections are then blended in the longitudinal direction by fitting parametric cubic-spline curves through coordinate points which define the conic sections in the cross-sectional planes. Both the cross-sectional and longitudinal curves may be modified by specifying particular segments as straight lines and slopes at selected points. Slopes may be continuous or discontinuous and finite or infinite.

After a satisfactory surface fit has been obtained, cards may be punched with the data necessary to form a geometry subroutine package for use in other computer programs. At any position on the body, coordinates, slopes and second partial derivatives are calculated.

The method is applied to a blunted  $70^\circ$  delta wing, and it was found to generate the geometry very well.

## INTRODUCTION

Many disciplines require a mathematical description of three-dimensional surfaces which cannot be represented by simple mathematical expressions. The location and slopes of points on a body are needed in the analysis of structures and inviscid flow fields. Viscous flow-field analyses require the body curvature in addition to location and slopes. The geometrical properties of simple shapes like spheres, cones, ellipsoids, and paraboloids can be described by relatively simple mathematical equations. However, many configurations of interest today, such as the space shuttle, are complex three-dimensional shapes whose geometry cannot be described easily. In many instances all the information that is given is a drawing with a plan view, a side view, and several cross sections of the vehicle. Sometimes models of a vehicle are available, and coordinate positions can be accurately measured on them. On the other hand, slopes cannot be accurately measured on a drawing or a model, and the determination of radii of curvature is even less accurate. This report develops a computer program which will surface fit mathematical relations to complex three-dimensional bodies. The method yields accurate coordinates and slopes and reasonably accurate radii of curvature at any position on the body.

Previous approaches to surface fitting three-dimensional bodies generally divided the surface into "patches" and represented each patch by flat surfaces, cubic or higher order polynomials, or conic sections (e.g., see refs. 1-3). Each of these methods has undesirable features in the forms used previously. Flat surfaces are completely unacceptable if radii of curvature or continuous slopes are needed. Cubic or higher order polynomials often lead to unwanted wiggles and bulges since they allow points of inflections. In ref. 3 this author applied the method of double splines, which used bi-cubic interpolation, to surface fit the coordinates of points in several cross-sectional planes of

three-dimensional bodies. However, it was found that bulges and/or dimples occurred in these surfaces, particularly when the thickness was much smaller than the span.

Reference 4 approximated the shape of a space shuttle orbiter with elliptical cross sections with different ellipticity on the windward and leeward sides. Cubic polynomials were used to define segments of the plan and thickness distributions. The coefficients of these polynomials were chosen to make the slopes continuous across boundaries of the longitudinal segments. However, points of inflections were found to occur inside the segments and thus gave the undesirable bumps, which in turn significantly affected the surface pressure and heating distributions calculated with this geometry.

Conic sections have been used to describe segments of aircraft contours for some time (see ref. 1). Both longitudinal and transverse contours were represented by conic sections, but the slopes at the ends of each segment had to be measured. As mentioned earlier, slopes are difficult to measure accurately. The present author developed a least-square curve-fitting technique (ref. 5) using general conic sections for each segment of a curve in a plane. Since all data points are not necessarily smooth, the curve was constrained to go through selected control points (which are boundaries for each conic segment) but fit the remaining data only in a least-square sense. In addition, the slope at each control point was constrained to be continuous. The results from this curve-fitting method were quite good. Another feature of conic sections is that they cannot yield curves with inflection points, and this feature can be a definite advantage. If an inflection point should be needed, then it could be made a control point with the resulting curves free of inflection points except at this control point.

Coons (ref. 2) developed a sophisticated technique to describe three-dimensional surfaces by using blending functions to blend the surface between the boundary

curves of each patch. A major difficulty in applying Coon's method is that the user must supply the coordinates, slopes, and twists (cross derivatives) at all four corners of each surface patch. This information is generally difficult to determine, particularly the cross derivatives (see ref. 6). Another disadvantage of using Coons' method in computer applications is that 64 parameters are needed to describe each surface patch, and for a large number of patches the amount of storage required by the computer may be excessive.

The method presented here for surface fitting three-dimensional bodies is somewhat similar to Coons' patching method, but it reduces the number of parameters required to describe a surface patch and is simpler to apply. Data points in cross sectional planes are curve-fit in a least-squares sense by segments of general conic sections. The conic sections are then blended in the longitudinal direction by fitting parametric splines through coordinate points which define the conic sections in the cross-sectional planes. This technique has the advantage of allowing the user to continually modify the cross-sectional curves and the longitudinal curves until the body shape has the desired features. Discontinuous slopes in both the circumferential and longitudinal directions may be specified.

After the body shape has been modified to the final form, the variables necessary to describe the shape may be punched on data cards. Then for applications of the method, a relatively small geometry subroutine can be used along with these data cards to calculate positions, slopes, and radii of curvature on three-dimensional bodies.



# SYMBOLS

$a_{m,i} : b_{m,j}$	parameters defined by eqs. (B5) - (B16)
$A_1, A_2, A_3$	coefficients of conic section given by eq. (18) in global coordinates.
$A_4, A_5$	
$A_j, B_j, C_j$	
$D_j, E_j, F_j$	coefficients of conic section given by eq. (19) in local coordinates.
$B$	
$B$	
$B$	parameter defined by eq. (26)
$d_p$	constant vector defined by eq. (20)
$f$	body radius defined by eq. (22)
$G_{pq}$	matrix defined by eq. (21)
$H$	parameter defined by eq. (27)
$K_j$	data point number of first control point in segment $j$
$m_j : n_j$	slopes defined by eqs. (5) and (6)
$N$	number of segments in a cross section
$P_j : Q_j$	terms defined by eqs. (A2) and (A3)
$r_{j,k}$	residual in eqs. (B1) and (B2)
$R_j$	term defined by eq. (A5)
$S$	chordal distance between coordinates
$T$	parameter defined by eq. (25)
$\bar{x}, \bar{y}, \bar{z}$	Cartesian coordinates, see Figure 2
$y, z$	local coordinates, see Figure 3
$\alpha_j, \beta_j, \gamma_j$	coefficients defined by eqs. (15), (16), and (17)
$\theta_j$	slope of segment $j$ , see Figure 5
$\Delta\theta_j$	$\theta_j - \theta_{j-1}$
$\phi$	circumferential angle defined by eq. (23)

## Subscripts

$h$	intermediate point in a cross-sectional segment
-----	---

j	segment number in a cross section
k	data point number in a cross section
r	reference point
s	slope point of a cross-sectional segment
o	first control point of a cross-sectional segment
l	last control point of a cross-sectional segment

Superscript

*	value where both $R_j = 0$ and $dR_j/dy = 0$
---	--

## ANALYSIS

For a typical application, the geometry of a body must be determined from a model, such as a wind tunnel model, or a three-view drawing. Generally several cross sections are obtainable from the model or drawing, and the coordinates of data points on the boundary of these cross sections can be measured or calculated. A three-dimensional surface must then be fit through all the body cross sections. Many times the data points in the cross sections are not completely smooth, and in those cases a smooth surface cannot be fit through all the data points. It is then desirable to have a smooth surface pass through designated data points, called control points, and pass close to, but not necessarily through, the other data points.

In the method presented here, the data points in each cross section are divided into segments and portions of general conic sections are curve-fit to the data points in each segment. The data points at the ends of each segment are designated as control points, and the curve is constrained to go through the control points as shown in Figure 1. A three-dimensional surface is then generated by "blending" the cross-sectional curves in the longitudinal direction. Consider first the technique for curve-fitting the data points in a cross-sectional plane.

### Curve-Fit in a Cross-Sectional Plane

As mentioned previously, the data points in a cross-sectional plane are generally not completely smooth, and in those cases a smooth curve cannot be made to pass through every data point. Therefore, the data points are divided into segments, and a portion of a general conic section is curve-fit in a least-squares sense through the data points in that segment. The curve is constrained to go through the control points (end points of a segment) and also have a

continuous slope at each control point unless the slope at a control point is specified otherwise.

Define a three-dimensional coordinate system  $\bar{x}, \bar{y}, \bar{z}$  with  $\bar{x}$  in the longitudinal direction and  $\bar{x} = \text{constant}$  is a cross-sectional plane (see Figure 2). Let  $j$  denote the segment number in a cross-sectional plane as shown in Figure 1, with the first segment starting on the positive  $\bar{y}$  - axis and  $j$  increasing clockwise. For each segment it is convenient to initially use a local coordinate system  $y, z$  with the origin on the first control point and the positive  $y$ -axis passing through the control point at the other end of the segment (see Figure 3). In this local coordinate system it is easy to investigate possibilities of complex roots and interpret the coefficients of a general conic section geometrically, whereas it is difficult to interpret them geometrically in the global coordinates  $\bar{y}, \bar{z}$ .

For the  $j^{\text{th}}$  segment the equation for a general conic section is given by (ref.7)

$$A_j y^2 + B_j yz + C_j z^2 + D_j y + E_j z + F_j = 0 \quad (1)$$

Only 5 of the 6 coefficients are independent since the equation may be divided by any non-zero coefficient. The constraints that the curve pass through the two control points

$$y = 0, z = 0 \quad \text{and} \quad y = y_j, z = 0 \quad \text{yield}$$

$$F_j = 0 \quad (2)$$

$$D_j = -A_j y_j \quad (3)$$

By differentiating eq. (1), the slope in local coordinates is given by

$$\frac{dz}{dy} = \frac{A_j y_j - 2A_j y - B_j z}{B_j y + 2C_j z + E_j} \quad (4)$$

Although the slopes at the ends of the segments are generally not known, define

$$m_j = \left( \frac{dz}{dy} \right) \quad \text{at } y = 0, z = 0 \quad (5)$$

and

$$n_j = \left( \frac{dz}{dy} \right) \quad \text{at } y = y_j, z = 0 \quad (6)$$

Using these two equations in eq. (4) it follows that

$$E_j = A_j y_j / m_j \quad (7)$$

and

$$B_j = -A_j \left( \frac{1}{m_j} + \frac{1}{n_j} \right) \quad (8)$$

These results show that the end slopes do not affect the coefficient  $C_j$ . On the other hand, the product  $A_j C_j$  determines the nature of the general conic section (ref. 7). If  $B_j^2 - 4A_j C_j = 0$  the conic is a parabola, if  $B_j^2 - 4A_j C_j < 0$  the conic is an ellipse, and if  $B_j^2 - 4A_j C_j > 0$  the conic is a hyperbola. Of particular interest is the possibility of complex roots when solving for  $z$  as a function of  $y$  in the region of interest. It is shown in Appendix A that  $z$  will have no complex roots in the region  $0 \leq y \leq y_j$  if

$$A_j C_j \geq (A_j / m_j) (A_j / n_j) \quad (9)$$

For prescribed slopes  $m_j, n_j$  it is interesting to observe how the product  $A_j C_j$  affects a conic section. (See Figure 4).

Unless a slope is specified at a control point, the method used here constrains the slope in global coordinates  $(\bar{y}, \bar{z})$  to be continuous at a control point. This makes the conic section in one segment dependent on data points in other segments as well as its own. From Figure 5 it can be seen that continuity of slope at

control point  $j$  requires

$$\tan^{-1} n_{j-1} = \tan^{-1} m_j + \Delta\theta_j \quad (10)$$

This equation can be expanded and rearranged in the form

$$\frac{A_{j-1}}{(A/n)_{j-1}} = \frac{(A/m)_j \sin \Delta\theta_j + A_j \cos \Delta\theta_j}{(A/m)_j \cos \Delta\theta_j - A_j \sin \Delta\theta_j} \quad (11)$$

This last equation is non-linear in the coefficients  $A_j$ , which would cause difficulty in obtaining a solution for them. However, as mentioned previously only 5 of 6 coefficients in a given segment are independent. This allows an additional constraint to be imposed without affecting the overall equation for the general conic section. The additional constraint used here is to equate separately the numerators and denominators of both sides of eq. (11). This gives two equations which are linear in the coefficients  $A_j$ , and which may be combined to yield

$$(A/n)_j = (A_j \cos \Delta\theta_{j+1} - A_{j+1}) / \sin \Delta\theta_{j+1} \quad (12)$$

$$(A/m)_j = (A_{j-1} - A_j \cos \Delta\theta_j) / \sin \Delta\theta_j \quad (13)$$

For the first segment ( $j = 1$ )  $m_1$  is specified and equation (13) is not needed, and also for the last segment ( $j = N$ )  $n_N$  is specified and eq. (12) is not needed.

Now substitute eqs. (2), (3), (7), (8), (12), and (13) into eq. (1) to obtain the equation for the conic sections as

$$\alpha_j A_{j-1} + \beta_j A_j + \gamma_j A_{j+1} + y_j^2 C_j = 0 \quad (14)$$

where

$$\alpha_j = \begin{cases} (zy_j - zy) / \sin \Delta \theta_j & \text{for } j > 1 \\ 0 & \text{for } j = 1 \end{cases} \quad (15)$$

$$\beta_j = \begin{cases} y^2 - (1/m_1 + \cot \Delta \theta_2)yz + y_j z / m_1 - y_j y & \text{for } j = 1 \\ y^2 + (\cot \Delta \theta_j - \cot \Delta \theta_{j+1})yz - \cot \Delta \theta_j y_j z - y_j y & \text{for } 1 < j < N \\ y^2 + (\cot \Delta \theta_N - 1/n_N)yz - \cot \Delta \theta_N y_N z - y_N y & \text{for } j = N \end{cases} \quad (16)$$

$$\gamma_j = \begin{cases} yz / \sin \Delta \theta_{j+1} & \text{for } j < N \\ 0 & \text{for } j = N \end{cases} \quad (17)$$

These results indicate that there are only two unknowns in each segment,  $A_j$  and  $C_j$ . However, there is one more segment than interior control points which means one of these coefficients is arbitrary. Here  $A_1 = 1$  is used arbitrarily (unless the conic section requires  $A_1 = 0$ ) and the independent unknowns become  $C_1, C_j, A_j$  ( $j = 2, \dots, N$ ). If there were as many data points as unknowns, and if some conic section could be made to pass through these data points, then eq. (14) could be applied to all the data points to give  $(2N-1)$  linear equations for the  $(2N-1)$  coefficients  $C_1, C_j, A_j$  for  $j = 2, \dots, N$ . However, there are generally more than  $(2N-1)$  data points, and if eq. (14) were applied to all of them an overdetermined system of linear equations (see ref. 8) would result. Therefore, a least-squares solution of the overdetermined system is used to determine the coefficients  $C_1, C_j, A_j$  for  $j = 2, \dots, N$ . This procedure is described in Appendix B. After obtaining the coefficients in this manner, the inequality of eq. (9) is checked for the possibility of complex roots for  $z$  in each segment. If the inequality of eq. (9) is not satisfied, the coefficient  $C_j$

in that section is replaced by the value obtained using the equality sign in eq. (9). As mentioned earlier, the coefficient  $C_j$  does not affect the slopes of the curve at the end points of the segment, and hence  $C_j$  does not affect other segments. The equality sign in eq. (9) gives 2 straight lines as shown in Figure 4.

In order to solve for  $z$  as a function of  $y$  from eq. (14), a quadratic equation must be solved and the proper choice of the + or - sign must be determined beforehand for each segment. It is shown in Appendix A that in order to make  $z \equiv 0$  at  $y = y_j$  (a control point at the end of segment  $j$ ), the + sign must be used if  $(A/m)_j > 0$  and  $(A/n)_j < 0$ , and the - sign must be used if  $(A/m)_j < 0$  and  $(A/n)_j > 0$ . \*

Once the coefficients  $A_j$  and  $C_j$  are determined, all the conic sections are completely defined for a given cross-sectional plane. In order to put these results into a form suitable for "blending" the cross sections in the longitudinal direction, the conic section for each segment is redefined in terms of 4 points - the two control points at the ends of the segment, a slope point which determines the slopes at the end points, and finally an intermediate point on the curve between the end points (see Figure 6). The 3 points on the curve and the two slopes at the end points of a segment are sufficient to determine new coefficients  $A_1, A_2, A_3, A_4$ , and  $A_5$  for the general conic section. \*\*

$$A_1 \bar{y}^2 + A_2 \bar{y}\bar{z} + A_3 \bar{z}^2 + A_4 \bar{y} + A_5 \bar{z} + 1 = 0 \quad (18)$$

---

\*Note that in order for a conic section to pass through the control points,  $n_j < 0$  if  $m_j > 0$  and  $n_j > 0$  if  $m_j < 0$ .

\*\*Note that the conic section given by eq. (18) must have the constant 1 replaced by 0 if the curve is to pass through the origin  $\bar{y} = 0, \bar{z} = 0$ .



in global coordinates  $\bar{y}, \bar{z}$ . (see ref. 1). This process is done for each segment in a cross section, and the 5 coefficients will become functions of the longitudinal coordinate  $\bar{x}$  when blending a segment of a cross section with corresponding segments in the other cross sections.

### Longitudinal Variation of Cross Section

In order to determine the longitudinal variation of the coefficients in eq. (18), a three-dimensional curve is fit through each of the 4 points used to define the conic section of corresponding segments (see Figure 7). In contrast to the cross sectional data points, these curves must pass through each of the points in the longitudinal direction. They are represented by their projections in the  $\bar{x}-\bar{y}$  and  $\bar{x}-\bar{z}$  planes; hence, two planar curves are used to represent each three-dimensional curve. The parametric method of cubic splines (ref. 9) is used to curve-fit each planar curve, with the chordal distance between the coordinate points as the parameter. The parametric spline allows infinite slopes whereas the regular spline will not. To facilitate the application of the splines, slopes in the longitudinal direction are calculated at each cross section from the parametric spline. Then for the region between two successive cross sections, the  $\bar{y}$  and  $\bar{z}$  coordinates of a three-dimensional curve are represented by cubic polynomials in  $\bar{x}$  with the coefficients determined by the coordinates and slopes at the two cross sections. When the slope of a coordinate at a cross section is infinite, then that coordinate is represented by a conic section for the region between that cross section and the one following it. If a longitudinal curve should not give the desired variation, slopes at selected cross sections may be modified by the user.

Consider now the longitudinal variation of a conic section. In each cross-sectional plane, eq. (18) will hold but the coefficients  $A_1, A_2, A_3, A_4$ , and  $A_5$  will vary with  $\bar{x}$ . As mentioned previously, these coefficients are determined

by 4 defining points (the two control points, an intermediate point, and the slope point). For each segment, the 5 equations used to determine the coefficients  $A_q$  ( $q=1, \dots, 5$ ) are formed by applying eq. (18) to the 3 points on the cross-sectional curve -- the two control points  $\bar{y}_0, \bar{z}_0$  and  $\bar{y}_1, \bar{z}_1$ , the intermediate point  $\bar{y}_h, \bar{z}_h$  -- and the slopes at the ends of the segment using the slope point  $\bar{y}_s, \bar{z}_s$  (see Figure 6). This procedure yields the following 5 equations:\*

$$\sum_{q=1}^5 G_{pq} A_q = d_p \quad p = 1, \dots, 5 \quad (19)$$

where

$$d_p = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -2 \\ -2 \end{bmatrix} \quad (20)$$

and

$$G_{pq} = \begin{bmatrix} \bar{y}_0^2 & \bar{y}_0 \bar{z}_0 & \bar{z}_0^2 & \bar{y}_0 & \bar{z}_0 \\ \bar{y}_1^2 & \bar{y}_1 \bar{z}_1 & \bar{z}_1^2 & \bar{y}_1 & \bar{z}_1 \\ \bar{y}_h^2 & \bar{y}_h \bar{z}_h & \bar{z}_h^2 & \bar{y}_h & \bar{z}_h \\ 2\bar{y}_0 \bar{y}_s & (\bar{y}_0 \bar{z}_s + \bar{y}_s \bar{z}_0) & 2\bar{z}_0 \bar{z}_s & (\bar{y}_0 + \bar{y}_s) & (\bar{z}_0 + \bar{z}_s) \\ 2\bar{y}_1 \bar{y}_s & (\bar{y}_1 \bar{z}_s + \bar{y}_s \bar{z}_1) & 2\bar{z}_1 \bar{z}_s & (\bar{y}_1 + \bar{y}_s) & (\bar{z}_1 + \bar{z}_s) \end{bmatrix} \quad (21)$$

---

\*The two equations for the slopes were combined with the first two equations ( $p = 1$  and  $2$ ) to obtain the last two equations ( $p = 4$  and  $5$ ).

At any longitudinal position, eq. (19) can be solved by any standard matrix inversion routine, such as SIMEQ on the CDC computer and SIMQ on the IBM computer, to determine the coefficients  $A_q$ . The derivatives  $dA_q/d\bar{x}$  and  $d^2A_q/d\bar{x}^2$  can be obtained by differentiating eq. (19) and successively solving the resulting system of linear equations. The elements of  $G_{pq}$  and their derivatives with respect to  $\bar{x}$  are obtained from the three-dimensional curves which were spline-fit through the 4 points used to define the conic section for that segment in each cross-sectional plane.

### Geometry in Polar Coordinates

For some applications it is convenient to express the geometry in polar coordinates  $\bar{x}$ ,  $f$ , and  $\phi$  where  $f = f(\bar{x}, \phi)$  is the radius in a cross section measured from a reference point  $\bar{y}_r$ ,  $\bar{z}_r$ , i.e.

$$f = [(\bar{y} - \bar{y}_r)^2 + (\bar{z} - \bar{z}_r)^2]^{1/2} \quad (22)$$

The reference point may be taken as  $\bar{y}_r = 0$ ,  $\bar{z}_r = 0$ , but in some cases  $f$  is not single valued unless a reference point is chosen off the  $\bar{x}$  axis. The angle  $\phi$  is given by (see Figure 8)

$$\phi = \tan^{-1}[(\bar{z} - \bar{z}_r)/(\bar{y} - \bar{y}_r)] , 0 \leq \phi \leq 2\pi \quad (23)$$

with  $\phi = 0$  corresponding to the line  $\bar{z} = \bar{z}_r$  and  $\bar{y} - \bar{y}_r > 0$ . Since  $\bar{y} - \bar{y}_r = f \cos \phi$  and  $\bar{z} - \bar{z}_r = f \sin \phi$ , eq. (18) in polar coordinates becomes

$$Tf^2 + Bf + H = 0 \quad (24)$$

where

$$T = A_1 \cos^2 \phi + A_2 \cos \phi \sin \phi + A_3 \sin^2 \phi \quad (25)$$

$$B = (2A_1 \bar{y}_r + 2A_2 \bar{z}_r + A_4) \cos \phi + (2A_2 \bar{y}_r + 2A_3 \bar{z}_r + A_5) \sin \phi \quad (26)$$

$$H = A_1 \bar{y}_r^2 + A_2 \bar{y}_r \bar{z}_r + A_3 \bar{z}_r^2 + A_4 \bar{y}_r + A_5 \bar{z}_r + 1 \quad (27)$$

Figure 8 illustrates the polar coordinates in a cross sectional plane.

When solving eq. (24) for  $f$  from the quadratic formula, the + or - sign must be determined beforehand. To determine it, apply eq. (24) to the control point at the end of the segment  $(j-1)$ , where  $f = f_j$  and  $\phi = \phi_j$  are known, to obtain

$$H = -T_j f_j^2 - B_j f_j \quad (28)$$

Substitute  $H$  from this equation into eq. (24) and solve the resultant for  $f$  from the quadratic formula to get

$$f = \frac{-B \pm [B^2 + 4T(T_j f_j^2 + B_j f_j)]^{1/2}}{2T} \quad \text{for } T \neq 0 \quad (29)$$

Now when  $T=T_j$  and  $B=B_j$ , eq. (29) gives  $f=f_j$  if the + sign is used when  $B_j + 2T_j f_j > 0$  and the - sign when  $B_j + 2T_j f_j < 0$ . At this longitudinal location, this same sign is applicable for any other value of  $\phi$  within the circumferential boundaries of this segment. In general, the sign must be determined at each longitudinal position. The derivatives  $\partial f / \partial \bar{x}$ ,  $\partial f / \partial \phi$ ,  $\partial^2 f / \partial \bar{x}^2$ ,  $\partial^2 f / \partial \phi^2$ ,  $\partial^2 f / \partial \bar{x} \partial \phi$ , and even higher derivatives, are obtained by differentiating eq. (24). Note, however, that the coefficients  $A_1$ ,  $A_2$ ,  $A_3$ ,  $A_4$ , and  $A_5$  are functions of  $\bar{x}$ . These coefficients and their derivatives are determined by the method described in the previous section.

#### COMPUTATIONAL ALGORITHM

Given the set of data points  $(\bar{y}_k, \bar{z}_k)$  in cross sectional planes at several longitudinal stations.

- (1) For each cross-sectional plane, divide the data points into segments so that a conic section can be curve-fit to the data points in each segment by the least-squares technique developed herein.
- (2) If the curves fit to the cross sectional data points are not satisfactory, modify them by one or more of the following methods: (a) define new boundaries (control points) for segments, (b) specify slope(s) at control point(s) (slopes may be finite or infinite and continuous or discontinuous);

- (c) specify selected segments as straight lines, (d) a specific conic section can be specified for a segment by prescribing the slopes at the ends of the segment and using only one datum point between the end control points.
- (3) Represent the conic section for each segment in a cross-sectional plane in terms of the two control points at the ends of the segment, an intermediate point, and the slope point (see Figure 6).
  - (4) For each point found in step (3), spline-fit a three-dimensional longitudinal curve through it and the corresponding points in other cross-sectional planes (see Figure 7).
  - (5) If the longitudinal curves are not satisfactory, modify them by one or more of the following methods: (a) specify slope(s) at longitudinal station(s) (slopes may be finite or infinite and continuous or discontinuous), (b) specify selected longitudinal segments as straight lines, (c) redefine the boundaries (control points) of the segments in the cross-sectional planes so that the points used for the longitudinal spline-fit form a smooth curve.
  - (6) The geometrical properties of the surface may be computed in polar coordinates at any position  $\bar{x}$ ,  $\phi$  by the following steps:
    - a) Locate  $\bar{x}$  between two consecutive longitudinal stations, and then locate the cross-sectional segment which contains  $\phi$ .
    - b) Use the spline function to calculate the coordinates, slopes, and second derivatives of the 4 longitudinal curves for this segment at  $\bar{x}$  (see Figure 7).
    - c) Calculate the coefficients  $A_p$  ( $p=1, \dots, 5$ ) of the conic section at this location by use of eq.(19). Determine the first and second derivatives of  $A_p$  with respect to  $\bar{x}$  from the first and second derivatives of eq. (19).

- d) Calculate the body radius  $f$  from eq. (24), and the derivatives  $\partial f/\partial \bar{x}$ ,  $\partial f/\partial \phi$ ,  $\partial^2 f/\partial \bar{x}^2$ ,  $\partial^2 f/\partial \phi^2$ ,  $\partial^2 f/\partial \bar{x} \partial \phi$ , and even higher derivatives from derivatives of eq. (24).
- (7) After a satisfactory surface fit has been obtained, the data which must be retained for a geometry subroutine package are the coordinates and longitudinal slopes of the longitudinal curves at those longitudinal stations where cross-sectional data points were given. Then, the geometrical properties of the surface can be calculated at any position by the method outlined in Step (6) above.

Appendix C describes a computer program written for the IBM 360/175 computer to perform this computational algorithm. This appendix also describes the input and output data for the program.

#### APPLICATION TO 70° DELTA WING

The surface fitting method developed herein is applied to the 70° slab delta wing shown in Figure 9. The computer program described in Appendix C was used to calculate the results, which are presented in Appendix D. This example was chosen because it illustrates many of the options available to modify the longitudinal curves and because the results can be compared with an exact solution.

Cross-sectional data are input at the 3 longitudinal stations shown in Figure 9 and, due to symmetry, only the first quadrant is used. Two segments (three control points) are needed to represent the cross section at  $\bar{x}=10$ . The first segment is a straight line and the second is one-fourth of an ellipse. The least-square curve-fit technique represents the ellipse exactly by specifying a zero slope at control point  $j=2$ , an infinite slope at control point  $j=3$ , and one datum point between these two control points. Although two segments must also be used for the other two cross sections, only one segment is necessary to specify the circle at  $\bar{x}=0.65798$  and the

ellipse at  $\bar{x}=1.0$ . Therefore, the first two data points, which are also control points, are made coincident. Then the exact curves are calculated from the least-square curve-fit by specifying a zero slope at control point  $j=2$ , an infinite slope at control point  $j=3$ , and one datum point between these two control points. The exact location of this datum point is irrelevant except that it must lie on the desired curve.

Recall that each three-dimensional longitudinal curve is represented by its projections in the  $\bar{x}-\bar{y}$  and  $\bar{x}-\bar{z}$  planes. Therefore, 16 longitudinal planar curves are used for this example (8 for each cross-sectional segment). Modifications are made to the initial spline fits to 14 of these curves since their exact shapes are readily obtained from Figure 9. The initial spline fit to the curves  $\bar{z}_1$  for segment 1 and  $\bar{y}_1$  for segment 2 are correct, and thus no modifications are necessary for these two longitudinal curves.

The geometrical properties  $f$  and its derivatives are listed in Appendix D for the circumferential angles  $\phi = 0, \pi/6, \pi/3$ , and  $\pi/2$  at  $\bar{x} = 0.3, 1.0, 2.0$ , and 5. The results at  $\bar{x}=1.0, 2.0$ , and 5.0 are exact (within the accuracy of single precision on the IBM 360/175 computer), whereas some inaccuracies are noted at  $\bar{x}=0.3$ . Here the cross section is a circle, thus the exact solution gives  $\partial f / \partial \phi = 0, \partial^2 f / \partial \phi^2 = 0, \partial^2 f / \partial \bar{x} \partial \phi = 0$ , and the derivatives  $\partial f / \partial \bar{x}$  and  $\partial^2 f / \partial \bar{x}^2$  should be independent of  $\phi$ . The surface fit could be made nearly exact at all positions by modifying the slopes of the longitudinal curves  $\bar{y}_h$  and  $\bar{z}_h$  for the second cross-sectional segment at  $\bar{x}=0.65798$ .

#### CONCLUDING REMARKS

An algorithm has been developed for surface fitting three-dimensional bodies. Data points in several cross-sectional planes of a body are required as input data. The data points in each cross section are then divided into segments, and a least-squares technique is used to curve fit portions of general conic sections to the

data points in each segment. The longitudinal variation of the cross sections is established by fitting parametric splines through the coordinates of points used to define the conic sections in the cross-sectional planes. If the surface fit is unsatisfactory, it can be modified by specifying slopes at the ends of segments or requiring selected segments to be straight lines. The surface geometry can best be analyzed through the use of an interactive computer graphics environment.

After a satisfactory surface fit to the body has been obtained, data cards can be punched and used with some of the subprograms to form a geometry subroutine package. This package is convenient for use in other computer programs requiring a mathematical model of the body geometry. It will calculate the body coordinates, slopes, and second derivatives. Computational time is short, and the amount of storage required is relatively small. The calculated data could also be used as input data for Coon's patching method (ref. 2), if the user should desire to employ that method in place of the one developed here.



## APPENDIX A

### Solution for Conic Section Equation

Substitute eqs. (2), (3), (7), and (8) into (1) to obtain the conic section in the form

$$C_j z^2 + P_j z + Q_j = 0 \quad (A1)$$

where

$$P_j = -[(A/m)_j + (A/n)_j]y + (A/m)_j y_j \quad (A2)$$

and

$$Q_j = A_j y(y - y_j) \quad (A3)$$

The solution of eq. (A1) is

$$z = (-P_j \pm R_j^{1/2}) / (2C_j) \quad \text{for } C_j \neq 0 \quad (A4)$$

where the discriminate is given by

$$R_j = P_j^2 - 4C_j Q_j \quad (A5)$$

In order to obtain real roots for  $z$  from eq. (A4), eq. (A5) must give

$$R_j \geq 0 \quad \text{for } 0 \leq y \leq y_j.$$

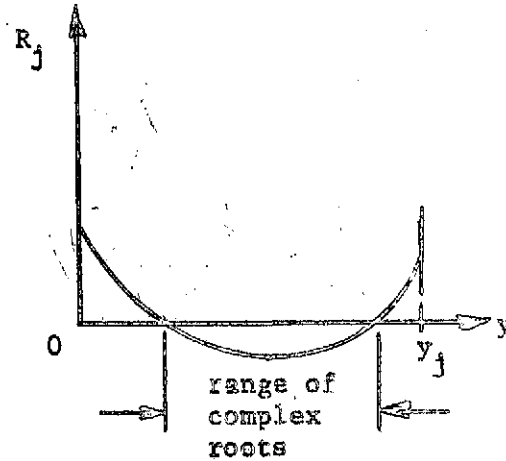
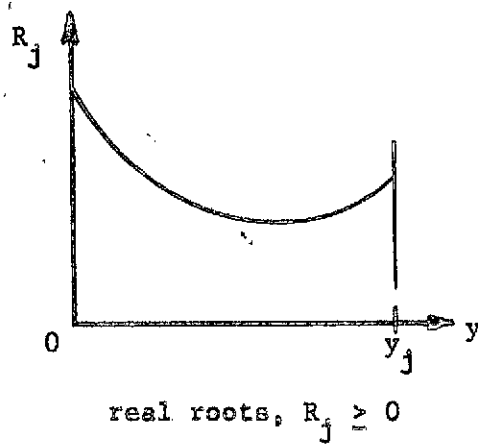
First, note that eq. (A5) gives  $R_j \geq 0$  when  $A_j C_j \geq 0$ . Also

$$R_j = (A/m)_j^2 y_j^2 \geq 0 \quad \text{at } y = 0$$

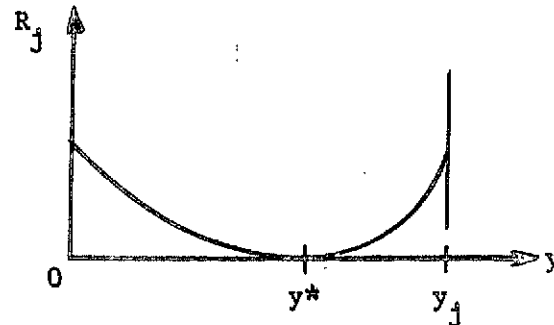
and

$$R_j = (A/n)_j^2 y_j^2 \geq 0 \quad \text{at } y = y_j$$

Thus the possibilities are illustrated below.



The minimum value of  $A_j C_j$  which still gives real roots corresponds to the case illustrated below



The minimum value of  $R_j$  occurs at  $y = y^*$  ( $0 \leq y^* \leq y_j$ ) where both  $R_j = 0$  and  $dR_j/dy = 0$ . The solution of  $R_j = 0$  and  $dR_j/dy = 0$ , using eq. (A5), yields the minimum value of  $A_j C_j$  as

$$(A_j C_j)^* = (A/m)_j (A/n)_j \quad (A6)$$

Thus eq. (A4) will yield real roots for  $z$  if

$$A_j C_j \geq (A/m)_j (A/n)_j \quad (A7)$$

The choice of the sign to be used in eq. (A4) is determined by requiring the equation to satisfy the coordinate of the control points  $z = 0, y = 0$

and  $z = 0$ ,  $y = y_j$ . For the first control point,  $y = 0$  in eq. (A4) gives

$$z = \frac{-(A/m)_j y_j + [(A/m)_j^2 y_j^2]^{1/2}}{2C_j} \quad (A8)$$

and then  $z = 0$  requires the + sign if  $(A/m)_j > 0$  and the - sign if  $(A/m)_j < 0$ .

At the second control point,  $y = y_j$  in eq. (A4) gives

$$z = \frac{(A/n)_j y_j + [(A/n)_j^2 y_j^2]^{1/2}}{2C_j} \quad (A9)$$

and  $z = 0$  requires the - sign if  $(A/n)_j > 0$  and the + sign if  $(A/n)_j < 0$ .

These conditions are all compatible because a conic section passing through the two control points will have  $n_j < 0$  if  $m_j > 0$  and  $n_j > 0$  if  $m_j < 0$ . The sign given by these conditions can be used in eq. (A4) for all values of  $y$  in the range  $0 \leq y \leq y_j$ .

## APPENDIX B

### Least-Squares Solution for Conic Section Coefficients

Eq. (14) cannot generally be satisfied at all data points because there would be more equations than unknowns. Accordingly, eq. (14) is applied at data point  $k$  and rewritten as

$$\alpha_{j,k} A_{j-1} + \beta_{j,k} A_j + \gamma_{j,k} A_{j+1} + y_k^2 C_j = r_{j,k} \quad (B1)$$

where  $\alpha_{j,k}$ ,  $\beta_{j,k}$ , and  $\gamma_{j,k}$  are the values of  $\alpha_j$ ,  $\beta_j$ , and  $\gamma_j$  evaluated at  $y = y_k$ ,  $z = z_k$  in segment  $j$ ; and  $r_{j,k}$  is called the residual. The least squares solution of the overdetermined system of equations determines the coefficients  $C_1$ ,  $C_j$ ,  $A_j$  ( $j=2, \dots, N$ ) which minimize the sum of the residuals squared (see ref. 8).

Define  $K_j$  as the data point number which corresponds to the first control point in segment  $j$ . Square eq. (B1) and sum over all the data points in continuous segments  $j=1, \dots, N$  to obtain

$$\sum_{j=1}^N \sum_{k=K_j}^{K_{j+1}} [\alpha_{j,k} A_{j-1} + \beta_{j,k} A_j + \gamma_{j,k} A_{j+1} + y_k^2 C_j]^2 = \sum_{j=1}^N \sum_{k=K_j}^{K_{j+1}} r_{j,k}^2 \quad (B2)$$

The right side of eq. (B2) is minimized by the system of equations obtained by setting partial derivatives of eq. (B2) with respect to the independent coefficients equal to zero. The result of setting partial derivatives with respect to  $A_2, \dots, A_N$  equal to zero gives the following system of equations:

$$\begin{aligned} & a_{m,1} A_{m-2} + a_{m,2} A_{m-1} + a_{m,3} A_m + a_{m,4} A_{m+1} \\ & + a_{m,5} A_{m+2} + a_{m,6} C_{m-1} + a_{m,7} C_m + a_{m,8} C_{m+1} = 0 \end{aligned} \quad (B3)$$

$m = 2, \dots, N$

The result of setting partial derivatives with respect to  $C_1, \dots, C_N$  equal to zero gives the following additional system of equations:

$$b_{m,1}A_{m-1} + b_{m,2}A_m + b_{m,3}A_{m+1} + b_{m,4}C_m = 0$$

$$m = 1, \dots, N \quad (B4)$$

The combined system of equations, i.e., eqs. (B3) and (B4), gives a system of  $(2N-1)$  linear equations for  $(2N-1)$  coefficients. The parameters used in eqs. (B3) and (B4) are defined as follows:

$$a_{m,1} = \sum_{k=K_{m-1}}^{K_m} \alpha_{m-1,k} \gamma_{m-1,k} \quad (B5)$$

$$a_{m,2} = \sum_{k=K_{m-1}}^{K_m} \beta_{m-1,k} \gamma_{m-1,k} + \sum_{k=K_m}^{K_{m+1}} \alpha_{m,k} \beta_{m,k} \quad (B6)$$

$$a_{m,3} = \sum_{k=K_{m-1}}^{K_m} \gamma_{m-1,k}^2 + \sum_{k=K_m}^{K_{m+1}} \beta_{m,k}^2 + \sum_{k=K_{m+1}}^{K_{m+2}} \alpha_{m+1,k}^2 \quad (B7)$$

$$a_{m,4} = \sum_{k=K_m}^{K_{m+1}} \beta_{m,k} \gamma_{m,k} + \sum_{k=K_{m+1}}^{K_{m+2}} \alpha_{m+1,k} \beta_{m+1,k} \quad (B8)$$

$$a_{m,5} = \sum_{k=K_{m+1}}^{K_{m+2}} \alpha_{m+1,k} \gamma_{m+1,k} \quad (B9)$$

$$a_{m,6} = \sum_{k=K_{m-1}}^{K_m} \gamma_{m-1,k}^2 \quad (B10)$$

$$a_{m,7} = \sum_{k=K_m}^{K_{m+1}} \beta_{m,k}^2 \quad (B11)$$

$$a_{m,8} = \sum_{k=K_{m+1}}^{K_{m+2}} y_k^2 \alpha_{m+1,k} \quad (B12)$$

$$b_{m,1} = a_{m-1,8} \quad (B13)$$

$$b_{m,2} = a_{m,7} \quad (B14)$$

$$b_{m,3} = a_{m+1,6} \quad (B15)$$

$$b_{m,4} = \sum_{k=K_m}^{K_{m+1}} y_k^4 \quad (B16)$$

Note that if slopes should be specified at selected control points, then the least-squares solution described above is applied to the segments inbetween two consecutive control points with specified slopes. The minimum number of data points (exclusive of the control points) that can be used in the least-squares solution is one in the first segment and two in each of the following segments. Any conic section can be prescribed for a segment by specifying the slopes at the ends of the segment and one datum point on the curve between the two control points. In this case eq. (B4) for  $m=1$  is the only equation to solve, and the least-squares technique determines the conic section which satisfies the specified conditions, provided a conic section can be fit through the prescribed slopes and datum point.

In the solution of the combined system of eqs. (B3) and (B4), the terms  $\alpha_{N+1,k}$  and  $\gamma_{N+1,k}$  must be interpreted as zero. Hence, the coefficients  $A_0$ ,  $A_{N+1}$ , and  $A_{N+2}$  do not appear in the resulting system, and recall that  $A_1 = 1$  unless the conic section should require  $A_1 = 0$ .

## APPENDIX C

### Description of Computer Program

Two computer programs are described here. The first determines the body geometry and allows modifications to be made to cross sectional and longitudinal curves until a satisfactory geometry is achieved. The second program uses data cards punched from the first program along with some of the subroutines to form a geometry subroutine package for use in other computer programs. Both computer programs are written in FORTRAN-IV for the IBM360/175 computer, but with a small number of changes they can be used on the CDC computers.

### Program for Determining Geometry

The following subprograms are called from the main program or its subprograms: GEOM, INGEO, LSTSQ, FZ, SPLX, SPLINE, FCN, DERIV, LINEQ, and SIMQ.

### Geometry Subroutine Package

After the Program for Determining Geometry has been modified to yield a satisfactory geometry, cards are punched by setting NP=1 in the input data (see Description of Input). Then these data cards along with subprograms GEOM, FCN, DERIV, LINEQ, and SIMQ form a geometry subroutine package which can be used in other computer programs for determining the body radius and its derivatives at prescribed locations  $\bar{x}$  and  $\phi$ . The statements necessary to use this geometry subroutine package in another computer program are given below under the listing of the Main Program for Geometry Subroutine Package.

### Main Program for Determining Geometry

This main program calls subroutine INGEO which, in turn, calculates all the parameters necessary to describe the geometry, and punches this information on cards if desired. The body radius  $R$  and its derivatives  $FX$ ,  $FP$ ,  $FXX$ ,  $FPP$ ,  $FXP$  are

calculated at prescribed locations  $\bar{x}$  and  $\phi$ , and the geometry is analyzed to see if it is satisfactory. Modifications are made to the geometry until it is satisfactory.

#### Main Program for Using Geometry Subroutine Package

The main program of some other computer program which uses this geometry subroutine package should have the COMMON and READ statements shown in the program listing. A CALL statement to calculate the body radius and its derivatives at prescribed values of  $\bar{x}$  and  $\phi$  is also listed.

#### Subroutine GEOM(XX, PH, YR, ZR, F, FX, FP, FXX, FPP, FXP)

At a given location  $(\bar{x}, \phi)$  and reference point  $(\bar{y}_r, \bar{z}_r)$ , this subroutine calculates the body radius and its derivatives. The arguments for this subroutine are:

XX, PH	$\bar{x}, \phi$
YR, ZR	$\bar{y}_r, \bar{z}_r$
F, FX, FP	$f, \partial f / \partial \bar{x}, \partial f / \partial \phi$
FXX, FPP, FXP	$\partial^2 f / \partial \bar{x}^2, \partial^2 f / \partial \phi^2, \partial^2 f / \partial \bar{x} \partial \phi$

Other program variables are:

CP, SP	$\cos \phi, \sin \phi$
T, B, H	parameters defined by eqs. (25), (26), and (27)
TX, TP	$\partial T / \partial \bar{x}, \partial T / \partial \phi$
TXX, TPP, TXP	$\partial^2 T / \partial \bar{x}^2, \partial^2 T / \partial \phi^2, \partial^2 T / \partial \bar{x} \partial \phi$

(similar definitions for BX, BP, etc.)

This subroutine is called from the Main Program.



### Subroutine INGE0

This subroutine reads part of the input data and calls other subroutines which read the remainder of the input data. When prescribed, it also punches cards with the data necessary for the Geometry Subroutine Package. Program variables are described in Description of Input and Output, and subroutines LSTSQ, SPLX, FCN, and DERIV. This subroutine is called from the Main Program for Determining Geometry.

### Subroutine LSTSQ (IT, NY)

This subroutine reads input data pertaining to a cross section and applies the least-squares technique described herein to determine the conic sections for each segment of the cross section. The dummy arguments for this subroutine are:

IT     longitudinal station number

NY     type of nose specified (see Description of Input)

The variables involved with the READ and WRITE statements are defined in the Descriptions of Input and Output. Other program variables are

A(I), B(I), C(I), D(I), E(I), GN(I)	}	See Description of Output
YY(IT, J, K)		

AM             $A_j/m_j$  in eq. (9)

AN             $A_j/n_j$  in eq. (9)

AMN            $(A_j/m_j)(A_j/n_j)$  in eq. (9)

AL, BET, GA    $\alpha_{j,k}$ ,  $\beta_{j,k}$ , and  $\gamma_{j,k}$  in eq. (B1)

BM(I), BZ(I)   Calculated  $\bar{y}$ ,  $\bar{z}$  corresponding to input data points

CAC            $A_j C_j$

G(I,J)        coefficient matrix for the system of eqs. (B3) and (B4) plus the equation  $A(1) = 1$ .

A(I)                    initially constant vector for the right side of the system of equations  $GX=A$ . After calling SIMQ the solution vector X is stored in A.

XC(J), YC(J)        $\bar{y}$  and  $\bar{z}$  coordinates, respectively, of control point J

ZM, ZN               slopes  $m_j$  and  $n_j$  in local coordinates

If inequality in eq. (9) is not satisfied, the message CAC LT AMN is printed along with the value of AMN and I. Then  $C_j$  is replaced by the value obtained using the equality sign in eq. (9).

This subroutine is called from subroutine INCEO only.

#### Function FZ(XX, I, JJ, XXB, YYB)

For a given cross-sectional segment and local coordinate y, this function calculates  $z(=FZ)$  in local coordinates and then transforms the point (y,z) to global coordinates ( $\bar{y}$ ,  $\bar{z}$ ). Eq. (A4) in Appendix A is used to calculate z. The arguments for this function are

XX                    value of local coordinate y

I                    segment identification no.

JJ                    segment number

XXB                   $\bar{y}$

YYB                   $\bar{z}$

This function is used in subroutine LSTSQ only.

#### Subroutine SPLX(NY,NJ)

This subroutine fits a cubic spline through longitudinal points to form longitudinal curves. The initial spline fits may be modified by specifying slopes or straight-line segments at selected longitudinal stations. See Description of Input Data for variables in READ statements. The arguments for this subroutine are:

NY                    nose shape (see Description of Input)

NJ                    number of longitudinal curves to be modified

Other program variables are

YXL(I,J,K)	left-handed slope	}    see Description of Output
YXR(I,J,K)	right-handed slope	

XP(I), YP(I)    parametric slopes  $\bar{dy}/dS$  and  $\bar{dz}/dS$  where S is the chordal distance between data points

This subroutine is called from subroutine INGE0 only.

Subroutine SPLINE (H,DE,K,AM,E,G, N, FP)

For an array of dependent variables, this subroutine solves the tridiagonal matrix system of algebraic equations to determine the coefficients and slopes for the cubic spline function as described in ref. 9. The dummy arguments for this subroutine are

H(J)                array representing difference between J+1 and J independent variables

DE(J)                array of dependent variables

K                    number of points,  $1 \leq J \leq K$

AM(J)                array of coefficients for spline function (second derivatives at each point)

E,G                  slope at the first and end points, respectively

N                    flag to describe end conditions of spline fit.

N = -2                indicates  $AM(2) = AM(1)$  and  $AM(K-1) = AM(K)$  are used in place of specifying E and G.

N = -1                indicates  $AM(1) = 0$  and  $AM(K) = 0$  are used in place of specifying E and G

N = 0                 indicates E is specified but  $AM(K-1) = AM(K)$  is used in place of specifying G

N = 1                 indicates both E and G are specified

This subroutine is called from subroutine SPLX only.

### Subroutine FCN(I,J,XX)

This subroutine calculates the  $\bar{y}$  and  $\bar{z}$  coordinates of 4 longitudinal stations by using a cubic variation in  $\bar{x}$  between longitudinal stations. Slopes and second derivatives of these coordinates are also calculated. The coefficients of the cubic variation are determined by the coordinates and slopes at the two longitudinal stations which surround the longitudinal position XX. If the slope at the longitudinal station just aft of XX is infinite, then the conic section

$$\Delta y^2 = A\Delta x + B\Delta x^2$$

is used for this segment, where

$$\Delta y = \bar{y} - \bar{y}(I-1) \quad \text{and} \quad \Delta x = \bar{x} - \bar{x}(I-1)$$

The coefficients A and B are determined by requiring the curve to pass through the coordinates at stations I and I-1 and have the prescribed slopes at these stations.

The arguments for this subroutine are

I                    first longitudinal station which exceeds XX  
J                    cross-sectional segment number  
XX                   longitudinal position  $\bar{x}$

Other program variables are

YXR(I,J,K), YXL(I,J,K)                    See Description of Output  
YY(I,J,K)                                    See Description of Output  
YL(L,M,N)                    parameters which are calculated at  $\bar{x} = XX$  where  
                                 L = 1 for  $\bar{y}$  and 2 for  $\bar{z}$   
                                 M = 1 for  $\bar{y}_0$  ( $\bar{z}_0$  if L = 2)  
                                 M = 2 for  $\bar{y}_1$  ( $\bar{z}_1$  if L = 2)  
                                 M = 3 for  $\bar{y}_h$  ( $\bar{z}_h$  if L = 2)  
                                 M = 4 for  $\bar{y}_s$  ( $\bar{z}_s$  if L = 2) (see Figure 7)  
                                 N = 1 for coordinate, 2 for derivative of coordinate  
                                                    respect to  $\bar{x}$ , and 3 for second derivative

This subroutine is called from subroutines INCEO and GEOM.

### Subroutine DERIV

This subroutine uses the parameters  $YL(L,M,N)$  calculated in subroutine FCN to calculate the coefficients  $A_p$  ( $p=1, \dots, 5$ ) and their longitudinal derivatives from eq. (19). Program variables are

AA(I)	initially $d_p$ as given by eq. (20); after calling SIMQ it becomes $A_p$ .
AX(I), AXX(I)	$dA_p/d\bar{x}$ , $d^2A_p/d\bar{x}^2$
G(I,J)	$G_{pq}$ (see eq. (21))
GS(I,J)	temporary storage for $G(I,J)$
GX(I,J), GXX(I,J)	$dG_{pq}/d\bar{x}$ , $d^2G_{pq}/d\bar{x}^2$
YL(L,M,N)	see description in subroutine FCN

This subroutine is called from subroutines INGE0 and GEOM. If the conic section should be a straight line, the determinant of  $G_{pq}$  is zero, and the solution is obtained from subroutine LINEQ.

### Subroutine LINEQ (G,AA)

This subroutine solves the two linear equations

$$G(1,4) X(1) + G(1,5) X(2) = AA(1)$$

$$G(2,4) X(1) + G(2,5) X(2) = AA(2)$$

for  $X(1)$  and  $X(2)$  and then stores the solution in  $AA(1)$  and  $AA(2)$ . The parameters  $AA(3)$ ,  $AA(4)$ , and  $AA(5)$  are set equal to zero. If the determinant of the coefficients of the two equations above should be zero, the message "DENOM IN LINEQ IS 0" is printed. This subroutine is called from subroutine DERIV whenever the conic section is a straight line.

### Subroutine SIMQ (G,A,N,K,KS)

This is one of the scientific subroutine packages in the IBM library to solve a system of linear equations  $GX=A$ . For the CDC computer, this subroutine may be

replaced by SIMEQ. The arguments for this subroutine are

G(I,J)	coefficient matrix in system $GX=A$ .
A(I)	vector A in the system $GX = A$ . On return to calling program, the solution vector X is stored in A.
N	the maximum order of G as stated in dimension statement of calling program.
K	the order of G; $1 \leq K \leq N$
KS	flag to indicate a solution is obtained or not. KS=1 indicates no solution.

## Appendix C (Continued)

### Description of Input

The input data for the computer program are described in the following steps which are also in the proper sequence.

Step 1.

```
      READ (1,100) NAM
100  FORMAT (20A3)
      :
      :
      READ (1,102) NY, NX, NCT, NJ, NP
102  FORMAT (5I5)
```

The read statements for this step appear in subroutine INGE0.

NAM      name used for body designation

NY      1 for blunt nose, 2 for sharp pointed nose, 3 for cut - off nose  
(i.e. the body radius is non - zero at the nose)

NX      number of longitudinal stations where cross sectional data are to  
be input. The nose station is included in NX although cross sectional  
data are not required there for blunted and sharp-pointed noses.

NCT      number of control points specified for all of the cross sections.  
Some control points may be made coincident if a cross section should  
require less than this number of control points.

NJ      number of longitudinal curves to be modified by specifying slopes  
or straight lines at selected longitudinal stations. Note that  
there are 8 longitudinal curves for each segment in a cross section  
since the 4 three-dimensional curves shown in Figure 7 are repre-  
sented by their projections in the  $\bar{x} - \bar{y}$  and  $\bar{x} - \bar{z}$  planes. Also,  
note that the longitudinal curve through control points is common to  
adjacent segments, and modifications made to it must be input for  
both segments.

NP        if cards are to be punched with the data necessary to use a geometry subroutine in some other computer program, NP=1. If cards are not to be punched NP $\neq$  1.

Step 2.

The 4 following statements are listed in subroutine INGEO.

```
      DO 2 I=1, NX
      READ (1,103) X(I)
103  FORMAT (F10.5)
      2  CALL LSTSQ(I,NY)
```

The statements below appear in subroutine LSTSQ.

```
      READ (1,101) N,NLT, NST, N2T
      READ (1,101) (NC(J), J=1, NCT)
101  FORMAT (16I5)
      IF (NLT.GT.0) READ (1,101) (NL(IL), IL=1, NLT)
      IF (N2T.GT.0) READ (1,100) (N2(I2), DX2(I2), DY2(I2), I2=1, N2T)
      IF (NST.GT.0) READ (1,100) (NS(IS), DXS(IS), DYS(IS), IS=1, NST)
100  FORMAT (I5, 2F10.5)
      READ (1,102) (XB(K), YB(K), K=1,N)
102  FORMAT (8F10.5)
      :
      :
      RETURN
```

The read statements in this step pertain to the information necessary to establish the curves in each cross sectional plane.

X(I)       value of  $\bar{x}$  at longitudinal station I, where I=1, ..., NX.

The information listed below is not required for the first longitudinal station (I=1) on blunted and sharp - pointed noses.

N        number of data points for this cross section. This number may vary



from one cross section to another.

NLT number of segments to be specified as straight lines.

NST number of control points where left - handed slopes are to be specified.

N2T number of control points where right - handed slopes are to be specified.

Note: The left - handed slope at a control point is defined as the slope, in global coordinates  $(\bar{y}, \bar{z})$ , of the segment which ends at the control point, whereas the right - handed slope is defined as the slope of the segment which begins at the control point. If the slope at a control point is continuous, the left - handed slope is the same as the right - handed slope. A left - handed slope may be specified at control point NS(IS) as  $d\bar{z}/d\bar{y} = DY(IS)/DXX(IS)$ , and this value is also used as the right - handed slope there unless a right - handed slope is specified or a straight - line segment begins there. A right - handed slope may be specified at control point N2(I2) as  $d\bar{z}/d\bar{y} = DY2(I2)/DX2(I2)$  provided a left - handed slope has already been specified at this same control point. Discontinuous slopes at the beginning or end of a straight - line segment are specified by  $d\bar{z}/d\bar{y} = DY(IS)/DXX(IS)$  at control point NS(IS). Slopes are specified in global coordinates  $(\bar{y}, \bar{z})$  by a numerator and a denominator so that zero and infinite slopes may be input. The first segment of a cross section (J=1) must be either a straight line or have the slope specified there [by  $d\bar{z}/d\bar{y} = DY(IS)/DXX(IS)$ ]. Also, the last segment (J = NCT - 1) must either end with a straight line or have the slope specified at the last control point (J = NCT) by  $d\bar{z}/d\bar{y} = DY(IS)/DXX(IS)$ .

NC(J) data point number to be designated a control point. The first and last data points must be control points, and also  $NC(J+1) \geq NC(J)$ .

NL(IL) number of a segment to be designated as a straight line.

N2(I2) control point number where a right - handed slope is specified.  
(see note above.)

DX2(I2) denominator of right - handed slope  $d\bar{z}/d\bar{y}$  at control point N2(I2).

DY2(I2) numerator of right - handed slope  $d\bar{z}/d\bar{y}$  at control point N2(I2).

NS(IS) control point number where a left - handed slope is specified.  
(see note above.)

DXX(IS) denominator of left - handed slope  $d\bar{z}/d\bar{y}$  at control point NS(IS).

DY(IS) numerator of left - handed slope  $d\bar{z}/d\bar{y}$  at control point NS(IS).

XB(K)  $\bar{y}$  coordinate of input data point K.

YB(K)  $\bar{z}$  coordinate of input data point K.

If this computer program is to be used with the body geometry expressed in polar coordinates, the first data point ( $K = 1$ ) must be on the line  $\phi = 0$  and succeeding data points should have  $\phi(K+1) \geq \phi(K)$  relative to the reference point  $\bar{y}_r, \bar{z}_r$  (see Figure 8).

The read statements in this step read data for all the cross sections before proceeding to Step 3.

### Step 3.

The following read statements are listed in subroutine SPLX and represent modifications to the longitudinal curves. If no longitudinal curves are to be modified from the initial spline fits, then  $NJ = 0$  should be input in Step 1 and the read statements in this step will be by - passed.

```

IF (NJ.EQ.0) RETURN
DO 17 IJK = 1, NJ
  READ (1,100) J,K, MLT, MST, M2T, MIT
100  FORMAT (16I5)
      IF (MLT.GT.0) READ (1,100) (ML(IL), IL = 1, MLT)
101  FORMAT (4(I10,F10.5))
      IF (M2T.GT.0) READ (1,101) (M2(I2), DY2(I2), I2 = 1, M2T)
      IF (MST.GT.0) READ (1,101) (MS(IS), DYX(IS), IS = 1, MST)
      IF (MIT.GT.0) READ (1,100) (MI(IT), IT = 1, MIT)
17  CONTINUE

```

J segment number in a cross - sectional plane where  $1 \leq J \leq (NCT - 1)$ .

K longitudinal curve to be modified in segment J; 1 for  $\bar{y}_0$ , 2 for  $\bar{y}_1$ ,  
 3 for  $\bar{y}_h$ , 4 for  $\bar{y}_s$ , 5 for  $\bar{z}_0$ , 6 for  $\bar{z}_1$ , 7 for  $\bar{z}_h$ , 8 for  $\bar{z}_s$ .  
 (see Figures 6 and 7).

MLT number of longitudinal segments to be specified as straight lines.

MST number of longitudinal points where finite, left - handed slopes  
 ( $d\bar{y}/d\bar{x}$  or  $d\bar{z}/d\bar{x}$ ) are specified.

M2T number of longitudinal points where finite, right - handed slopes  
 are specified.

Note: Comments in the note after the description of N2T in Step 2 apply  
 here also, except MS(IS) replaces NS(IS), M2(I2) replaces N2(I2), and  
 the slopes are ( $d\bar{y}/d\bar{x}$  or  $d\bar{z}/d\bar{x}$ ) = DYX(IS) at MS(IS) and ( $d\bar{y}/d\bar{x}$  or  $d\bar{z}/d\bar{x}$ )  
 = DY2(I2) at M2(I2). However, infinite slopes are input separately here.

MIT number of longitudinal stations where infinite slopes in the longi-  
 tudinal direction are specified.

ML(IL) longitudinal station number where the beginning of a straight line  
 segment is specified. The straight line will terminate at the next

longitudinal station.

M2(I2) longitudinal station number where a right - handed slope is specified (see note above)

DY2(I2) right - handed slope ( $\bar{dy}/\bar{dx}$  or  $\bar{dz}/\bar{dx}$ ) at M2(I2).

MS(IS) longitudinal station number where a left - handed slope is specified (see note above).

DYX(IS) left - handed slope ( $\bar{dy}/\bar{dx}$  or  $\bar{dz}/\bar{dx}$ ) at MS(IS).

MI(IT) longitudinal station number where the right - handed slope is specified as infinite.

## Appendix C (Continued)

### Description of Output

The data described in Step 1 of the Description of Input are printed using the same nomenclature. For each cross-sectional plane, the data described in Step 2 of the Description of Input are printed with the same nomenclature except YB and ZB are the  $\bar{y}$  and  $\bar{z}$  coordinates of the input data points. Additional output for each cross section are:

A,B,C,D,E	coefficients of conic section given by eq. (1)
SGN	sign to be used in eq. (A4)
SL0, SL1	slope $dz/dy$ in local coordinates at the beginning and end of a segment, respectively.
YB(CALC) ZB(CALC)	$\bar{y}$ and $\bar{z}$ corresponding to input data points, but calculated from conic section equation.
YY(I,J,K)	Coordinates of four points used to define conic section in cross-sectional segment J at longitudinal station X(I). K = 1 for $\bar{y}_0$ , 2 for $\bar{y}_1$ , 3 for $\bar{y}_h$ , 4 for $\bar{y}_s$ , 5 for $\bar{z}_0$ , 6 for $\bar{z}_1$ , 7 for $\bar{z}_h$ , 8 for $\bar{z}_s$ (see Figure 6)

The data described above are printed for each cross sectional plane. Then the longitudinal slopes computed at each cross section by the method of splines (before any modifications are made) are printed, where

I	longitudinal station number
J	cross-sectional segment number
K	same as K described in YY(I,J,K) above
YXL(I,J,K,)	left - handed longitudinal slope. ( $d\bar{y}/d\bar{x}$ or $d\bar{z}/d\bar{x}$ )
YXR(I,J,K)	right - handed longitudinal slope.

If modifications are made to the initial spline-fit for the longitudinal curves, the data in Step 3 of Description of Input are printed with the same nomenclature, and then the modified left - handed (YXL) and right - handed (YXR) slopes are printed.

Next, the coefficients for eq. (18) are printed for each longitudinal station I and cross-sectional segment J.

In the example used here, the following data are calculated and printed for prescribed longitudinal ( $\bar{x}$ ) and circumferential positions ( $\phi$ ) relative to the reference point  $\bar{y}_r, \bar{z}_r$ :

X, PHI	$\bar{x}, \phi$
F, FX, FP	$f, \partial f / \partial \bar{x}, \partial f / \partial \phi$
FXX, FPP, FXP	$\partial^2 f / \partial x^2, \partial^2 f / \partial \phi^2, \partial^2 f / \partial \bar{x} \partial \phi$

Appendix C (Continued)

Program Listing  
from IBM 360/175 Computer

Contents	Page
Main Program for Determining Geometry . . . . .	44
Main Program for Using Geometry Subroutine Package . . . . .	44
Subroutine INGE0 . . . . .	44
Subroutine SPLX . . . . .	45
Subroutine SPLINE . . . . .	48
Subroutine GEOM . . . . .	49
Subroutine FCN . . . . .	51
Subroutine DERIV . . . . .	51
Subroutine LINEQ . . . . .	53
Subroutine LSTSQ . . . . .	53
Function FZ . . . . .	58

```

$JOB      NCS.MAE.B2630/DEJARNETTE,TIME=30,PAGES=40,LINES=54
C  MAIN PROGRAM FOR DETERMINING GEOMETRY
C  EXAMPLE - 70 DEG DELTA WING
      WRITE(3,666)
666  FORMAT(1H1)
      CALL INGED
      WRITE(3,100)
100  FORMAT(///,17X,1HX,14X,3HPHI,14X,1HF,15X,2HFX,/,18X,2HFP,13X,3HFX
1,14X,3HFPP,13X,3HFXP)
      DO 2 I=1,4
        X=.3
        IF(I.EQ.2)X=1.
        IF(I.EQ.3)X=2.
        IF(I.EQ.4)X=5.
      DO 2 J=1,4
        PH=(J-1)*3.141592/6
        CALL GEOM(X,PH,0.,0.,F,FX,FP,FXX,FPP,FXP)
2  WRITE(3,101)X,PH,F,FX,FP,FXX,FPP,FXP
101  FORMAT(/,8X,4E16.7,/,9X,4E16.7)
      STOP
      END

```

```

C  HAIN PROGRAM FOR USING GEOMETRY SUBROUTINE PACKAGE
C  EXAMPLE - 70 DEG DELTA WING
      COMMON/COM2/X(20),YY(20,10,8),YXR(20,10,8),YXL(20,10,8),YL(2,4,3),
      INX,NCT
      READ(1,301)NX,NCT,NR
301  FORMAT(3I5)
      READ(1,300)(X(I),I=1,NX)
300  FORMAT(5E16.7)
      READ(1,300)((YY(I,J,K),K=1,8),J=1,NR),I=1,NX)
      READ(1,300)((YXL(I,J,K),K=1,8),J=1,NR),I=2,NX)
      NZ=NX-1
      READ(1,300)((YXR(I,J,K),K=1,8),J=1,NR),I=1,NZ)
      WRITE(3,100)
100  FORMAT(///,17X,1HX,14X,3HPHI,14X,1HF,15X,2HFX,/,18X,2HFP,13X,3HFX
1,14X,3HFPP,13X,3HFXP)
      DO 2 I=1,4
        XX=.3
        IF(I.EQ.2)XX=1.
        IF(I.EQ.3)XX=2.
        IF(I.EQ.4)XX=5.
      DO 2 J=1,4
        PH=(J-1)*3.141592/6
        CALL GEOM(X,PH,0.,0.,F,FX,FP,FXX,FPP,FXP)
2  WRITE(3,101)XX,PH,F,FX,FP,FXX,FPP,FXP
101  FORMAT(/,8X,4E16.7,/,9X,4E16.7)
      RETURN
      END

```

```

      SUBROUTINE INGED
C  READ INPUT DATA AND CALCULATE COEFFICIENTS
C  NY= 1 FOR BLUNT NOSE, 2 FOR SHARP POINTED NOSE, 3 FOR CUT OFF NOSE

```



```

C NX=NO. OF LONGITUDINAL STATIONS, NCT IS NO. OF CONTROL PTS., NJ= NO.
C OF LONGITUDINAL LINES TO BE MODIFIED
C NP=1 FOR PUNCHING DATA CARDS
COMMON/COM2/X(20),YY(20,10,8),YXR(20,10,8),YXL(20,10,8),YL(2,4,3),
1NX,NCT
COMMON/COM3/A(5),AX(5),AXX(5)
DIMENSION NAM(20)
READ(1,100)NAM
100 FORMAT(20A3)
WRITE(3,101)NAM
101 FORMAT(4X,20A3)
READ(1,102)NY,NX,NCT,NJ,NP
102 FORMAT(5I5)
WRITE(3,201) NY,NX,NCT,NJ,NP
201 FORMAT(/,8X, 3HNY=,I5,3X,3HNX=,I5,3X,4HNCT=,I5,3X,3HNJ=,I5,3X,3HN
1P=,I5)
DO 2 I=1,NX
READ(1,103)X(I)
103 FORMAT(F10.5)
2 CALL LSTSQ(I,NY)
CALL SPLX(NY,NJ)
WRITE(3,202)
202 FORMAT(/,11X,1HI,3X,1HJ,8X,4HA(1),12X,4HA(2),12X,4HA(3),/,41X,
1 4HA(4),12X,4HA(5))
IB=1
IF(NY.LE.2)IB=2
NR=NCT-1
DO 3 I=IB,NX
DO 3 J=1,NR
CALL FCN(I,J,X(I))
IF(YL(1,1,1).EQ.YL(1,2,1).AND.YL(2,1,1).EQ.YL(2,2,1))GO TO 3
CALL DERIV
WRITE(3,203)I,J,A(1),A(2),A(3),A(4),A(5)
3 CONTINUE
203 FORMAT(/,8X,2I4,3E16.7,/,33X,2E16.7)
IF (NP.NE.1) GO TO 4
WRITE(2,301)NX,NCT,NR
301 FORMAT(3I5)
WRITE(2,300)(X(I),I=1,NX)
300 FORMAT(5E16.7)
WRITE(2,300)((YY(I,J,K),K=1,8),J=1,NR),I=1,NX)
WRITE(2,300)((YXL(I,J,K),K=1,8),J=1,NR),I=2,NX)
NZ=NX-1
WRITE(2,300)((YXR(I,J,K),K=1,8),J=1,NR),I=1,NZ)
4 RETURN
END

SUBROUTINE SPLX(NY,NJ)
COMMON/COM2/X(20),YY(20,10,8),YXR(20,10,8),YXL(20,10,8),YL(2,4,3),
1N,NCT
DIMENSION Y(20),YP(20),XP(20),DS(20),YM(20),XM(20),ML(5),M2(5),DY2
1(5),MS(5),DYX(5),MI(5),XX(20)
NR=N-1
NQ=NCT-1

```

```

DO 2 J=1,NQ
DO 2 K=1,8
Y(1)=YY(1,J,K)
DO 3 I=2,N
Y(I)=YY(I,J,K)
T1=X(I)-X(I-1)
T2=YY(I,J,K)-YY(I-1,J,K)
3 DS(I)=SQRT(T1**2+T2**2)
IF(NY.GE.2)GO TO 4
YP(1)=0
IF(YY(2,J,K).NE.0)YP(1)=DS(2)/YY(2,J,K)
XP(1)=0
CALL SPLINE(DS,Y,N,YM,YP(1),1.,0,YP)
CALL SPLINE(DS,X,N,XM,XP(1),1.,0,XP)
YXR(1,J,K)=1.E20
IF(YY(2,J,K).EQ.0)YXR(1,J,K)=0
IB=2
GO TO 31
4 CALL SPLINE(DS,Y,N,YM,1.,1.,-2,YP)
CALL SPLINE(DS,X,N,XM,1.,1.,-2,XP)
IB=1
31 DO 2 I=IB,N
YXL(I,J,K)=YP(I)/XP(I)
2 YXR(I,J,K)=YXL(I,J,K)
WRITE(3,203)
203 FORMAT(/,11X,1HI,3X,1HJ,3X,1HK,5X,10HYXL(I,J,K),6X,10HYXR(I,J,K),
13X,18HFOR INITIAL SPLINE)
DO 32 J=1,NQ
WRITE(3,401)
401 FORMAT(5X)
DO 32 K=1,8
WRITE(3,558)
558 FORMAT(5X)
DO 32 I=1,N
32 WRITE(3,204) I,J,K,YXL(I,J,K),YXR(I,J,K)
204 FORMAT(8X,3I4,2E16.7)
IF(NJ.EQ.0)RETURN
DO 17 IJK=1,NJ
ML(1)=0
MS(1)=0
M2(1)=0
MI(1)=0
READ(1,100)J,K,MLT,MST,M2T,MIT
100 FORMAT(16I5)
WRITE(3,205) J,K,MLT,MST,M2T,MIT
205 FORMAT(/,8X,2HJ=,I3,3X,2HK=,I3,3X,4HMLT=,I3,3X,4HMST=,I3,3X,
1 4HM2T=,I3,3X,4HMIT=,I3)
IF(MLT.GT.0)READ(1,100)(ML(IL),IL=1,MLT)
101 FORMAT(4(I10,F10.5))
IF(M2T.GT.0)READ(1,101)(M2(I2),I2=1,M2T)
IF(MST.GT.0)READ(1,101)(MS(IS),IS=1,MST)
IF(MIT.GT.0)READ(1,100)(MI(IT),IT=1,MIT)
IF(M2T.GT.0)WRITE(3,206)(M2(I2),I2=1,M2T)
IF(MST.GT.0)WRITE(3,207)(MS(IS),IS=1,MST)

```

```

      IF(MIT.GT.0)WRITE(3,208)(MI(IT),IT=1,MIT)
      IF(MLT.GT.0)WRITE(3,209)(ML(IL),IL=1,MLT)
206  FORMAT(//,10X,2HM2,7X,6HDY2/DX,/,(/,8X,14,E16.7))
207  FORMAT(//,10X,2HMS,7X,5HDY/DX,/,(/,8X,14,E16.7))
208  FORMAT(/ ,8X,3HMI=,514)
209  FORMAT(/ ,8X,3HML=,514)
      IS=1
      IL=1
      IZ=1
      IT=1
      IB=1
      DO 5 I=1,NR
      DYY=YY(I+1,J,K)-YY(I,J,K)
      DXX=X(I+1)-X(I)
      DSS=SQRT(DXX**2+DYY**2)
      IF(I.EQ.ML(IL))GO TO 6
      IF(I.EQ.MI(IT))GO TO 7
      IF(I.EQ.M2(IZ))GO TO 8
      IF(I.EQ.MS(IS))GO TO 30
      IF(KA.EQ.1)GO TO 9
      GO TO 5
9     MB=1
      MB1=I+1
      KA=0
      DO 10 IZ=MB1,N
      ME=IZ
      M=ME-MB+1
      DYE=YY(IZ,J,K)-YY(IZ-1,J,K)
      DXE=X(IZ)-X(IZ-1)
      DSE=SQRT(DYE**2+DXE**2)
      IF(IZ.EQ.MS(IS))GO TO 11
      IF(IZ.EQ.ML(IL))GO TO 12
10    CONTINUE
6     YXR(I,J,K)=DYY/DXX
      YXL(I+1,J,K)=DYY/DXX
      IF(I.GE.NR)GO TO 5
      DXE=X(I+2)-X(I+1)
      DYE=YY(I+2,J,K)-YY(I+1,J,K)
      DSE=SQRT(DXE**2+DYE**2)
      XP(1)=DSE/(DXE+DYY/DXX*DYE)
      YP(1)=XP(1)*DYY/DXX
      YXR(I+1,J,K)=DYY/DXX
      KA=1
      IF(IL.LT.MLT)IL=IL+1
      IF(I.EQ.MS(IS).AND.IS.LT.MST)IS=IS+1
      GO TO 5
7     YP(1)=DSS/DYY
      XP(1)=0
      YXR(I,J,K)=1.E20
      IF(IT.LT.MIT)IT=IT+1
      IF(I.EQ.MS(IS).AND.IS.LT.MST)IS=IS+1
      IB=2
      GO TO 9
8     XP(1)=DSS/(DXX+DY2(IZ)*DYY)

```

```

      YP(1)=DY2(I2)*XP(1)
      YXR(I,J,K)=YP(1)/XP(1)
      IF(I2.LT.M2T) I2=I2+1
      IF(I5.LT.MST) I5=I5+1
      GO TO 9
30    XP(1)=DSS/(DXK+DYX(I5)*DYY)
      YP(1)=DYX(I5)*XP(1)
      YXR(I,J,K)=YP(1)/XP(1)
      IF(I5.LT.MST) I5=I5+1
      GO TO 9
11    XP(M)=DSE/(DXE+DYX(I5)*DYE)
      YP(M)=DYX(I5)*XP(M)
      GO TO 13
12    FPB=(YY(I2+1,J,K)-YY(I2,J,K))/(X(I2+1)-X(I2))
      XP(M)=DSE/(DXE+FPB*DYE)
      YP(M)=FPB*XP(M)
13    DO 14 IQ=MB,I2
      IR=IQ-MB+1
      Y(IR)=YY(IQ,J,K)
      XX(IR)=X(IQ)
      IF(IR.EQ.1) GO TO 14
      T1=X(IQ)-X(IQ-1)
      T2=YY(IQ,J,K)-YY(IQ-1,J,K)
      DS(IR)=SQRT(T1**2+T2**2)
14    CONTINUE
      IB=1
      IF(M.LE.2) GO TO 16
      CALL SPLINE(DS,Y,M,YH,YP(1),YP(M),1,YP)
      CALL SPLINE(DS,XX,M,XM,XP(1),XP(M),1,XP)
16    DO 15 IQ=2,M
      JQ=IQ-MB-1
      YXL(JQ,J,K)=YP(IQ)/XP(IQ)
15    YXR(JQ,J,K)=YP(IQ)/XP(IQ)
5    CONTINUE
      WRITE(3,402) J,K,J,K
402   FORMAT(//,11X,1H1,4X,6HYXL(1,,I2,1H,,I2,1H),5X,6HYXR(1,,I2,
1H,,I2,1H),/)
      DO 403 IQ=1,N
403   WRITE(3,210) IQ,YXL(IQ,J,K),YXR(IQ,J,K)
210   FORMAT( 8X,14,2E16.7)
17    CONTINUE
      RETURN
      END

```

```

SUBROUTINE SPLINE(H,DE,K,AM,E,G,N,FP)
  DIMENSION M(K),DE(K),AM(K),A(20),B(20),C(20),D(20),Q(20),U(20),
1P(20),FP(20)

```

C E IS LEFT DERV, G IS RIGHT DERV, N=1 FOR E & G SPECIFIED,  
C N=0 FOR E SPECIFIED, N=-1 FOR NEITHER SPECIFIED AND M=0 AT BOTH ENDS  
C N=-2 FOR M AT END POINTS = M NEXT TO END POINTS

```

      A(1)=0.
      B(1)=1.
      C(1)=.5
      IF(N.EQ.-1) C(1)=0.

```

```

      IF(N.EQ.-2)C(1)=-1.
      D(1)=3.*((DE(2)-DE(1))/H(2)-E)/H(2)
      IF(N.LT.0)D(1)=0.
      A(K)=.5
      IF(N.EQ.-1)A(K)=0.
      B(K)=1.
      IF(N.EQ.0)A(K)=-B(K)
      IF(N.EQ.-2)A(K)=-B(K)
      C(K)=0.
      D(K)=3.*(G-(DE(K)-DE(K-1))/H(K))/H(K)
      IF(N.LE.0)D(K)=0
      Q(1)=-C(1)
      U(1)=D(1)
      P(1)=B(1)
      DO 20 J=2,K
      IF(J.EQ.K)GO TO 21
      B(J)=1.
      C(J)=H(J+1)/(H(J)+H(J+1))/2.
      A(J)=.5-C(J)
      D(J)=3.*(((DE(J+1)-DE(J))/H(J+1))-(DE(J)-DE(J-1))/H(J))/(H(J)+H(J+
1))
21 P(J)=A(J)*Q(J-1)+B(J)
   Q(J)=-C(J)/P(J)
20 U(J)=(D(J)-A(J)*U(J-1))/P(J)
   AM(K)=U(K)
   KM1=K-1
   DO 22 J=1,KM1
   JR=K-J
   AM(JR)=Q(JR)*AM(JR+1)+U(JR)
22 FP(JR+1)=(AM(JR+1)/3+AM(JR)/6)*H(JR+1)+(DE(JR+1)-DE(JR))/H(JR+1)
   FP(1)=(-AM(1)/3-AM(2)/6)*H(2)+(DE(2)-DE(1))/H(2)
   RETURN
   END

      SUBROUTINE GEOM(XX,PH,YR,ZR,F,FX,FP,FXX,FPP,FXP)
      COMMON/COM2/X(20),YY(20,10,8),YXR(20,10,8),YXL(20,10,8),YL(2,4,3),
1IX,NCT
      COMMON/COM3/AA(5),AX(5),AXX(5)
C   LOCATE I AND J FOR X AND PHI
      JX=NCT-1
      DO 2 I=2,IX
      IF(XX.LT.X(I))GO TO 3
2   CONTINUE
      I=IX
3   DO 4 J=1,JX
      CALL FCN(I,J,XX)
      DYJ=YL(1,2,1)-YR
      DZJ=YL(2,2,1)-ZR
      FJ=SQRT(DYJ**2+DZJ**2)
      PHJ=ARCOS(DYJ/FJ)
      IF(DZJ.LT.0)PHJ=2.0*3.141592-PHJ
4   CONTINUE
      J=JX
5   CALL DERIV

```

```

CP=DYJ/FJ
SP=DZJ/FJ
H=AA(1)*YR**2+AA(2)*YR*ZR+AA(3)*ZR**2+AA(4)*YR+AA(5)*ZR+1.0
B1=2.0*AA(1)*YR+2.0*AA(2)*ZR+AA(4)
B2=2.0*AA(2)*YR+2.0*AA(3)*ZR+AA(5)
C2=CP**2
S2=SP**2
S2P=2*SP*CP
C2P=C2-S2
TT=AA(1)*C2+AA(2)*S2P/2+AA(3)*S2
B=B1*CP+B2*SP
T3=B+2*TT*FJ
SGN=-1
IF(T3.GT.0)SGN=1
CP=COS(PH)
SP=SIN(PH)
C2=CP**2
S2=SP**2
S2P=2*SP*CP
C2P=C2-S2
T=AA(1)*C2+AA(2)*S2P/2+AA(3)*S2
B=B1*CP+B2*SP
TX=AX(1)*C2+AX(2)*S2P/2+AX(3)*S2
TXX=AXX(1)*C2+AXX(2)*S2P/2+AXX(3)*S2
TP=(AA(3)-AA(1))*S2P+AA(2)*C2P
TPP=2*((AA(3)-AA(1))*C2P-AA(2)*S2P)
TXP=(AX(3)-AX(1))*S2P+AX(2)*C2P
B1X=2.0*AX(1)*YR+2.0*AX(2)*ZR+AX(4)
B2X=2.0*AX(2)*YR+2.0*AX(3)*ZR+AX(5)
BX=B1X*CP+B2X*SP
B1XX=2.0*AXX(1)*YR+2.0*AXX(2)*ZR+AXX(4)
B2XX=2.0*AXX(2)*YR+2.0*AXX(3)*ZR+AXX(5)
BXX=B1XX*CP+B2XX*SP
BP=-B1*SP+B2*CP
BPP=-B
BPX=-B1X*SP+B2X*CP
T2=B**2-4.0*T*H
TS=0
IF(T2.GT.0)TS=SQRT(T2)
IF(ABS(T).GT.1.E-09)GO TO 9
IF(B.NE.0)F=-1/B
GO TO 7
F=(-B+SGN*TS)/2/T
DE=2*F*T+B
F2=F*F
FP=(-F2*TP-F*BP)/DE
FX=(-F2*TX-F*BX)/DE
FPP=(-2*F*TP**2*Y-4*F*FP*TP-F2*TPP-2*F*BP-F*BP)/DE
FXX=(-2*F*TX**2*Y-4*F*FX*TX-F2*TX**2*Y-2*F*BX-F*BX)/DE
UP=-F2*TXP-2*F*FP*TX-2*F*FP*TX
UN=-2*F*FP*FX-F*BPX-FX*BP-FP*BX
FXP=(UP+UN)/DE
RETURN
END

```

```

      SUBROUTINE FCN(I,J,XX)
C   CALCULATE X-VARIATION FROM SPLINES
C   IN YY(I,J,K), I AND J ARE GEOMETRIC POSITIONS K= 1 - Y0, 2 - Y1,
C   3 - YH, 4 - YS, 5 THRU 8 SAME FOR Z
C   YL(L,M,N), L=1 FOR Y, 2 FOR Z, M = 1 FOR Y0, 2 FOR Y1, 3 FOR YH, 4 FOR Y
C   N= 1 FOR F, 2 FOR FX, 3 FOR FXX
      COMMON/COM2/X(20),YY(20,10,8),YXR(20,10,8),YXL(20,10,8),YL(2,4,3),
      IIX,NCT
      DIMENSION F(3,3)
      DX=X(I)-X(I-1)
      U=(XX-X(I-1))/DX
      U2=U**2
      U3=U**3
      F(1,1)=2*U3-3*U2+1
      F(2,1)=(U3-2*U2+U)*DX
      F(3,1)=(U3-U2)*DX
      F(1,2)=(6*U2-6*U)/DX
      F(1,3)=(12*U-6)/DX**2
      F(2,2)=(3*U2-4*U+1)
      F(2,3)=(6*U-4)/DX
      F(3,2)=3*U2-2*U
      F(3,3)=(6*U-2)/DX
      DO 2 L=1,2
      DO 2 M=1,4
      K=4*(L-1)+M
      IF(YXR(I-1,J,K).GE.1.E20)GO TO 3
      DO 5 N=1,3
      KK=1
      IF(N.GE.2)KK=0
      5 YL(L,M,N)=(YY(I-1,J,K)-YY(I,J,K))*F(1,N)+YY(I,J,K)*KK+YXR(I-1,J,K)
      1*F(2,N)+YXL(I,J,K)*F(3,N)
      GO TO 2
C   SEGMENT BEGINNING WITH INFINITE SLOPE
  3   T1=YY(I,J,K)-YY(I-1,J,K)
      T=T1/DX
      A=2.0*T1*(T-YXL(I,J,K))
      B=T*(2.0*YXL(I,J,K)-T)
      SGN=1.0
      V=XX-X(I-1)
      IF(T1.LT.0)SGN=-1.0
      DY=SGN*SQRT(A*V+B*V**2)
      YL(L,M,1)=YY(I-1,J,K)+DY
      IF(DY.NE.0) GO TO 4
      YL(L,M,2)=0.0
      YL(L,M,3)=0.0
      GO TO 2
  4   YL(L,M,2)=(A+2.0*8*DX)/2.0/DY
      YL(L,M,3)=(B-YL(L,M,2)**2)/DY
  2   CONTINUE
      RETURN
      END

```

```

      SUBROUTINE DERIV
C   CALCULATE A(1),A(2),A(3),A(4),A(5) AND THEIR DERIVATIVES W,R,T,X

```

```

COMMON/COM2/X(20),YY(20,10,8),YXR(20,10,8),YXL(20,10,8),YL(2,4,3),
IN,NCT
COMMON/COM3/AA(5),AX(5),AXX(5)
DIMENSION Q(5),QX(5),QXX(5),G(5,5),GX(5,5),GXX(5,5),GS(5,5)
DO 2 J=4,5
J3=J-3
Q(J)=YL(J3,4,1)
QX(J)=YL(J3,4,2)
QXX(J)=YL(J3,4,3)
DO 2 I=1,3
G(I,J)=YL(J3,I,1)
GX(I,J)=YL(J3,I,2)
2. GXX(I,J)=YL(J3,I,3)
K=4
L=4
DO 3 J=1,3
IF(J.EQ.3)K=5
IF(J.GE.2)L=5
DO 3 I=1,3
G(I,J)=G(I,K)*G(I,L)
GX(I,J)=G(I,K)*GX(I,L)+GX(I,K)*G(I,L)
3. GXX(I,J)=G(I,K)*GXX(I,L)+2*GX(I,K)*GX(I,L)+GXX(I,K)*G(I,L)
DO 4 I=4,5
I3=I-3
G(I,2)=G(I3,5)*Q(4)+G(I3,4)*Q(5)
GX(I,2)=G(I3,5)*QX(4)+GX(I3,5)*Q(4)+G(I3,4)*QX(5)+GX(I3,4)*Q(5)
T=G(I3,5)*QXX(4)+2*GX(I3,5)*QX(4)+GXX(I3,5)*Q(4)
GXX(I,2)=T+G(I3,4)*QXX(5)+2*GX(I3,4)*QX(5)+GXX(I3,4)*Q(5)
DO 4 J=4,5
K=1
IF(J.EQ.5)K=3
G(I,K)=2*G(I3,J)*Q(J)
GX(I,K)=2*(G(I3,J)*QX(J)+GX(I3,J)*Q(J))
GXX(I,K)=2*(G(I3,J)*QXX(J)+2*GX(I3,J)*QX(J)+GXX(I3,J)*Q(J))
G(I,J)=G(I3,J)+Q(J)
GX(I,J)=GX(I3,J)+QX(J)
4. GXX(I,J)=GXX(I3,J)+QXX(J)
DO 5 I=1,3
5. AA(I)=-1
AA(4)=-2
AA(5)=-2
DO 6 I=1,5
DO 6 J=1,5
6. GS(I,J)=G(I,J)
CALL SIMQ(GS,AA,5,5,KS)
IF(KS.EQ.1)GO TO 9
DO 7 I=1,5
AX(I)=0
DO 7 J=1,5
AX(I)=AX(I)-AA(J)*GX(I,J)
7. GS(I,J)=G(I,J)
CALL SIMQ(GS,AX,5,5,KS)
IF(KS.EQ.1)GO TO 9
DO 8 I=1,5

```



```

      AXX(I)=0
      DO 8 J=1,5
      AXX(I)=AXX(I)-AA(J)*GXX(I,J)-2*AX(J)*GX(I,J)
8    GS(I,J)=G(I,J)
      CALL SIMQ(GS,AXX,5,5,KS)
      IF(KS.EQ.1) GO TO 9
      RETURN
9    CONTINUE
      AA(1)=-1
      AA(2)=-1
      CALL LINEQ(G,AA)
      AX(1)=-AA(4)*GX(1,4)-AA(5)*GX(1,5)
      AX(2)=-AA(4)*GX(2,4)-AA(5)*GX(2,5)
      CALL LINEQ(G,AX)
      AXX(1)=-AA(4)*GXX(1,4)-2*AX(4)*GX(1,4)-AA(5)*GXX(1,5)-2*AX(5)*GX
1(1,5)
      AXX(2)=-AA(4)*GXX(2,4)-2*AX(4)*GX(2,4)-AA(5)*GXX(2,5)-2*AX(5)*GX
1(2,5)
      CALL LINEQ(G,AXX)
      RETURN
      END

```

```

      SUBROUTINE LINEQ(G,AA)
      DIMENSION G(5,5),AA(5)
      DEN=G(1,4)*G(2,5)-G(1,5)*G(2,4)
      IF(DEN.EQ.0)GO TO 3
      UP4=AA(1)*G(2,5)-G(1,5)*AA(2)
      UP5=G(1,4)*AA(2)-AA(1)*G(2,4)
      AA(4)=UP4/DEN
      AA(5)=UP5/DEN
      DO 2 I=1,3
2    AA(I)=0
      RETURN
3    WRITE(3,202)
202  FORMAT(///,2X,19HDENOM IN LINEQ IS 0,/)
      RETURN
      END

```

```

      SUBROUTINE LSTSQ(IT,NY)
C   FOR THIS SUBROUTINE Z IS USED FOR X, X FOR Y, AND Y FOR Z
      COMMON/COM2/Z(20),YY(20,10,8),YXR(20,10,8),YXL(20,10,8),YL(2,4,3),
1IX,NCT
      DIMENSION NC(10),XB(20),YB(20),DX(10),SDA(10),CDA(10),CT(10),AG(10,
1),BG(10),AB(10),GG(10),BB(10),AA(11),GE(10),BE(10),AE(10),EE(10),X
1(20),Y(20),G(20,20),BZ(20),XS(10),YS(10),XH(10),YH(10),BM(20)
      DIMENSION NL(5),NS(5),DXX(5),DY(5),N2(5),DX2(5),DY2(5)
      COMMON/COM1/XC(10),YC(10),SA(10),CA(10),SGN(10),A(20),B(10),C(10),
1D(10),E(10)
      M=NCT-1
      IF(IT.EQ.1.AND.NY.LE.2)GO TO 32
      READ(1,101)N,NLT,NST,N2T
      READ(1,101)(NC(J),J=1,NCT)
101  FORMAT(16I5)
      NL(1)=0

```

```

N2(1)=0
IF(NLT.GT.0)READ(1,101)(NL(I),I=1,NLT)
IF(N2T.GT.0)READ(1,100)(N2(I2),DX2(I2),DY2(I2),I2=1,N2T)
IF(NST.GT.0) READ(1,100)(NS(IS),DXX(IS),DY(IS),IS=1,NST)
100 FORMAT(15,2F10.5)
READ(1,102)(XB(K),YB(K),K=1,N)
102 FORMAT(8F10.5)
C N=TOTAL NO. OF DATA PTS.,NL(I) IS NO. OF CONTROL PT. WHICH IS BEG. OF
C A ST. LINE SEG.,NLT IS TOTAL NO. OF ST. LINE SEGS.,SIMILARILY
C NS(I) AND NST ARE FOR INPUT SLOPES (MUST HAVE INITIAL AND END SLOPES
C INPUT OR HAVE A ST. LINE THERE)
WRITE(3,400)
400 FORMAT(/,8X,50H*****
WRITE(3,211) Z(IT)
WRITE(3,212) N,NLT,NST,N2T
WRITE(3,213)(NC(I),I=1,NCT)
IF(NLT.GT.0)WRITE(3,214)(NL(I),I=1,NLT)
IF(NST.GT.0)WRITE(3,215)(NS(I),DXX(I),DY(I),I=1,NST)
IF(N2T.GT.0)WRITE(3,216)(N2(I),DX2(I),DY2(I),I=1,N2T)
WRITE(3,217)(I,XB(I),YB(I),I=1,N)
211 FORMAT(/,8X,3HX =,E16.7)
212 FORMAT(/,8X,2HN=,I3,3X,4HNLT=,I3,3X,4HNST=,I3,3X,4HN2T=,I3)
213 FORMAT(/,8X,3HNC=,10I5)
214 FORMAT(/,8X,3HNL=,5I5)
215 FORMAT(/,10X,2HNS,8X,2HDY,13X,2HDZ,/, (8X,14,2E16.7))
216 FORMAT(/,10X,2HN2,7X,3HDY2,13X,3HDZ2,/, (8X,14,2E16.7))
217 FORMAT(/,8X,5HPT.NO,7X,2HYB,14X,2HZB,/, (8X,15,2E16.7))
DO 11 J=1,NCT
K=NC(J)
XC(J)=XB(K)
11 YC(J)=YB(K)
WRITE(3,300)
300 FORMAT(/,11X,1HJ,8X,4HA(J),12X,4HB(J),12X,4HC(J),12X,4HD(J),/,21X
1,4HE(J),12X,6HSGN(J),10X,6HSL0(J),10X,6HSL1(J))
IS=1
IL=1
MZ=M
KA=0
I2=1
DO 17 IQ=1,MZ
IF(IQ.EQ.NL(IL))GO TO 8
IF(IQ.EQ.N2(I2))GO TO 19
IF(IQ.EQ.NS(IS))GO TO 10
IF(KA.EQ.1)GO TO 15
GO TO 17
15 MB=IQ
MB1=MB+1
KA=0
DO 9 IZ=MB1,NCT
IF(IZ.EQ.NS(IS))GO TO 16
IF(IZ.EQ.NL(IL))GO TO 12
9 CONTINUE
8 BY1=YC(IQ+1)-YC(IQ)
DX1=XC(IQ+1)-XC(IQ)

```

```

YY(IT,IQ,1)=XC(IQ)
YY(IT,IQ,2)=XC(IQ+1)
YY(IT,IQ,3)=(XC(IQ)+XC(IQ+1))/2
YY(IT,IQ,4)=YY(IT,IQ,3)
YY(IT,IQ,5)=YC(IQ)
YY(IT,IQ,6)=YC(IQ+1)
YY(IT,IQ,7)=(YC(IQ+1)+YC(IQ))/2
YY(IT,IQ,8)=YY(IT,IQ,7)
IF(IL.LT.NLT) IL=IL+1
IF(IL.LT.1) IL=1
IF(IQ.EQ.NS(IS).AND.IS.LT.NST) IS=IS+1
KA=1
A(IQ)=0
B(IQ)=0
C(IQ)=0
D(IQ)=0
E(IQ)=1
SGN(IQ)=1
ZM=0
ZN=0
WRITE(3,301) IQ,A(IQ),B(IQ),C(IQ),D(IQ),E(IQ),SGN(IQ),ZM,ZN
K=NC(IQ)
L=NC(IQ+1)-1
DSB=SQRT(DY1**2+DX1**2)
IF(DSB.LE.1.E-20) DSB=1
CAA=DX1/DSB
SAA=DY1/DSB
DO 33 JJ=K,L
XHL=(XB(JJ)-XB(K))*CAA+(YB(JJ)-YB(K))*SAA
BM(JJ)=XC(IQ)+XHL*CAA
33 BZ(JJ)=YC(IQ)+XHL*SAA
GO TO 17
10 DY1=DY(IS)
DX1=DXX(IS)
IF(IS.LT.NST) IS=IS+1
GO TO 15
19 DY1=DY2(I2)
DX1=DX2(I2)
IF(I2.LT.N2T) I2=I2+1
IF(IS.LT.NST) IS=IS+1
GO TO 15
12 ME=I2
DYN=YC(I2+1)-YC(I2)
DXN=XC(I2+1)-XC(I2)
M=ME-MB
GO TO 7
16 ME=I2
DYN=DY(IS)
DXN=DXX(IS)
M=ME-MB
7 DO 14 I=1,20
A(I)=0
DO 14 J=1,20
14 G(I,J)=0

```

```

DO 2 I=1,M
K=NC(I+MB-1)
L=NC(I+MB)
T1=XB(L)-XB(K)
T2=YB(L)-YB(K)
DX(I)=SQRT(T1**2+T2**2)
SA(I)=T2/DX(I)
CA(I)=T1/DX(I)
IF(I.EQ.1)GO TO 2
SDA(I)=SA(I)*CA(I-1)-CA(I)*SA(I-1)
CDA(I)=CA(I)*CA(I-1)+SA(I)*SA(I-1)
CT(I)=CDA(I)/SDA(I)
2 CONTINUE
YP=(DY1*CA(1)-DX1*SA(1))/(DX1*CA(1)+DY1*SA(1))
YPN=(DYN*CA(M)-DXN*SA(M))/(DXN*CA(M)+DYN*SA(M))
IF(YP.EQ.0.OR.YPN.EQ.0)GO TO 8
CT(1)=-1/YP
CT(M+1)=1/YPN
SDA(M+1)=1
SDA(1)=1
DO 3 I=1,M
AG(I)=0
BG(I)=0
AB(I)=0
GG(I)=0
BB(I)=0
AA(I)=0
GE(I)=0
BE(I)=0
AE(I)=0
EE(I)=0
K=NC(I+MB-1)
L=NC(I+MB)-1
DO 3 J=K,L
X(J)=(XB(J)-XB(K))*CA(I)+(YB(J)-YB(K))*SA(I)
Y(J)=-((XB(J)-XB(K))*SA(I)+(YB(J)-YB(K))*CA(I))
AL=Y(J)*(DX(I)-X(J))/SDA(I)
BET=X(J)**2+(CT(I)-CT(I+1))*X(J)*Y(J)-CT(I)*DX(I)*Y(J)-DX(I)*X(J)
GA=X(J)*Y(J)/SDA(I+1)
AG(I)=AG(I)+AL*GA
BG(I)=BG(I)+BET*GA
AB(I)=AB(I)+AL*BET
GG(I)=GG(I)+GA*GA
BB(I)=BB(I)+BET**2
AA(I)=AA(I)+AL**2
GE(I)=GE(I)+GA*Y(J)**2
BE(I)=BE(I)+BET*Y(J)**2
AE(I)=AE(I)+AL*Y(J)**2
EE(I)=EE(I)+Y(J)**4
3 CONTINUE
AA(M+1)=0
G(1,1)=1
A(1)=1
IF(M.EQ.1)GO TO 13

```

```

DO 4 I=2,M
  IF(I.GE.3)G(I,I-2)=AG(I-1)
  G(I,I-1)=BG(I-1)+AB(I)
  G(I,I)=GG(I-1)+BB(I)+AA(I+1)
  IF(I.LE.(M-1))G(I,I+1)=BG(I)+AB(I+1)
  IF(I.LE.(M-2))G(I,I+2)=AG(I+1)
  IM=I+M
  G(I,IM-1)=GE(I-1)
  G(I,IM)=BE(I)
  IF(I.LT.M)G(I,IM+1)=AE(I+1)
  G(IM,I-1)=AE(I)
  G(IM,I)=BE(I)
  IF(I.LT.M)G(IM,I+1)=GE(I)
4 G(IM,IM)=EE(I)
13 G(M+1,1)=BE(1)
  G(M+1,2)=GE(1)
  G(M+1,M+1)=EE(1)
  I2M=2*M
  DO 23 I=1,I2M
    DO 22 J=1,I2M
      IF(G(I,J).NE.0)GO TO 23
22 CONTINUE
      IF(M.EQ.1)GO TO 25
      IZ=I+MB-1
      IF(I.GT.M)IZ=IZ-M
      IF(NLT.LT.IL)GO TO 24
      DO 21 KI=IL,NLT
21 NL(KI+1)=NL(KI)
24 NLT=NLT+1
      NL(IL)=IZ
      IF(IZ.EQ.IQ)GO TO 27
      GO TO 12
25 A(1)=1
      A(2)=1/YP/YPN
      GO TO 26
27 NL(IL)=IQ+1
      GO TO 8
23 CONTINUE
      CALL SIMQ(G,A,20,I2M,KS)
      IF(KS.EQ.1) WRITE(3,777)
777 FORMAT(///,8X,18HNO SOLN FOR MATRIX)
26 AM=1/YP
      DO 5 I=1,M
        J=I+MB-1
        IF(I.EQ.1)GO TO 6
        AM=(A(I-1)-CDA(I)*A(I))/SDA(I)
6 AN=A(M)/YPN
        IF(I.LT.M)AN=(A(I)*CDA(I+1)-A(I+1))/SDA(I+1)
        CAC=A(M+1)*A(I)
        AMN=AN*AM
        IF(CAC.LT.AMN)WRITE(3,218) CAC,AMN,I
218 FORMAT(///,8X,10HCAC LT AMN,3X,4HCAC=,E16.7,4X,4HAMN=,E16.7,4X,
1 2HI=,I3,/)
        IF(CAC.LT.AMN)A(M+1)=AMN/A(I)

```

```

C(I)=A(M+I)
B(I)=- (AM+AN)
D(I)=-A(I)*DX(I)
E(I)=AM*DX(I)
SGN(I)=1
IF(AM.LT.0)SGN(I)=-1
ZN=A(I)/AM
ZM=A(I)/AN
XSL=-ZN*DX(I)/(ZM-ZN)
YSL=ZM*XSL
XS(I)=XC(J)+XSL*CA(I)-YSL*SA(I)
YS(I)=YC(J)+XSL*SA(I)+YSL*CA(I)
VHL=FZ(XSL,I,J,XH(I),YH(I))
YY(IT,J,1)=XC(J)
YY(IT,J,2)=XC(J+1)
YY(IT,J,3)=XH(I)
YY(IT,J,4)=XS(I)
YY(IT,J,5)=YC(J)
YY(IT,J,6)=YC(J+1)
YY(IT,J,7)=YH(I)
YY(IT,J,8)=YS(I)
301 WRITE(3,301) J,A(I),B(I),C(I),D(I),E(I),SGN(I),ZN,ZM
FORMAT(/,8X,I4,4E16.7,/,13X,E16.7,F16.5,2E16.7)
K=NC(I+MB-1)
L=NC(I+MB)-1
DO 5 JJ=K,L
VHL=FZ(X(JJ),I,J,BH(JJ),BZ(JJ))
5 CONTINUE
17 CONTINUE
BH(N)=XB(N)
BZ(N)=YB(N)
WRITE(3,219)(I,BH(I),BZ(I),I=1,N)
219 FORMAT(/,8X,5HPT.NO,5X,8HYB(CALC),8X,8HZB(CALC),/, (8X,I5,2E16.7))
WRITE(3,220)IT,IT,IT,IT,IT,IT,IT,IT
220 FORMAT(/,11X,1HJ,5X,3HYY(,I2,5H,J,1),6X,3HYY(,I2,5H,J,2),6X,3HYY(
I,I2,5H,J,3),6X,3HYY(,I2,5H,J,4),/,18X,3HYY(,I2,5H,J,5),6X,3HYY(,
I,I2,5H,J,6),6X,3HYY(,I2,5H,J,7),6X,3HYY(,I2,5H,J,8))
DO 20 I=1,MZ
20 WRITE(3,221) I,(YY(IT,I,J),J=1,8)
221 FORMAT(/,8X,I4,4E16.7,/,13X,4E16.7)
RETURN
32 DO 31 I=1,M
DO 31 J=1,8
31 YY(IT,I,J)=0
RETURN
END

```

```

FUNCTION FZ(XX,I,JJ,XXB,YYB)
COMMON/CON1/XC(10),YC(10),SA(10),CA(10),SGN(10),A(20),B(10),C(10),
ID(10),E(10)
T1=B(I)*XX+E(I)
T2=XX*(A(I)*XX+D(I))
IF(C(I).EQ.0)GO TO 2
T3=4*T2*C(I)

```

```

T4=T1**2-T3
IF(T4.LT.0)GO TO 3
FZ=(-T1+SGN(I)*SQRT(T4))/C(I)/2
7 XXB=XC(JJ)+XX*CA(I)-FZ*SA(I)
  YYB=YC(JJ)+XX*SA(I)+FZ*CA(I)
  RETURN
2 FZ=-T2/T1
  GO TO 7
3 WRITE(3,222) T4
222 FORMAT(///,8X,15HDISCRIMINATE IS,E16.7,3X,14HFOR FZ, SET =0)
  FZ=-T1/C(I)/2
  GO TO 7
END

```

## APPENDIX D

## Input and Output for 70° Delta Wing

Input

70 DEG DELTA WING

Step 1

1	4	3	14				
0.							
.65798							
4	1	2					
1	2	4					
1							
2	0.	1.					
3	1.	0.					
.9396926	0.			.9396926	0.	.664463	.664463 0.
1.							
4	1	2					
1	2	4					
1							
2	0.	1.					
3	1.	0.					
1.	0.			1.	0.	.5	.921605 0.
10.							1.064178
4	1	2					
1	2	4					
1							
2	0.	1.					
3	1.	0.					
1.	0.			1.	3.27573	.5	4.197335 0.
							4.33991

Step 2

Step 3

1	1	1	0	0	1
3					
1					
1	2	1	0	0	1
3					
2					
1	3	1	0	0	1
3					
1					



Input (Continued)

1	4	1	0	0	1
3					
1					
1	6	3			
1	2	3			
1	7	3			
1	2	3			
1	8	3			
1	2	3			
2	1	1	0	0	1
3					
1					
2	3	1	0	0	1
3					
1					
2	4	1	0	0	1
3					
1					
2	5	3			
1	2	3			
2	6	2	0	0	1
2	3				
1					
2	7	1	0	0	1
3					
1					
2	8	2	0	0	1
2	3				
1					

# Appendix D (Continued)

## Output

70 DEG DELTA WING

NY= 1 NX= 4 NCT= 3 NJ= 14 NP= 0

\*\*\*\*\*

X = 0.6579800E 00

N= 4 NLT= 1 NST= 2 N2T= 0

NC= 1 2 4

NL= 1

NS	DY	DZ
2	0.0000000E 00	0.1000000E 01
3	0.1000000E 01	0.0000000E 00

PT.NO	YB	ZB
1	0.9396926E 00	0.0000000E 00
2	0.9396926E 00	0.0000000E 00
3	0.6644630E 00	0.6644630E 00
4	0.0000000E 00	0.9396926E 00

J	A(J) E(J)	B(J) SGN(J)	C(J) SLO(J)	D(J) SL1(J)
1	0.0000000E 00 0.1000000E 01	0.0000000E 00 1.00000	0.0000000E 00 0.0000000E 00	0.0000000E 00 0.0000000E 00
2	0.1000000E 01 -0.1328925E 01	-0.0000000E 00 -1.00000	0.9999935E 00 -0.1000000E 01	-0.1328925E 01 0.1000000E 01

PT.NO	YB(CALC)	ZB(CALC)
1	0.9396926E 00	0.0000000E 00
2	0.9396923E 00	-0.3371769E-06
3	0.6644623E 00	0.6644634E 00
4	0.0000000E 00	0.9396926E 00

J	YY( 2,J,1) YY( 2,J,5)	YY( 2,J,2) YY( 2,J,6)	YY( 2,J,3) YY( 2,J,7)	YY( 2,J,4) YY( 2,J,8)
1	0.9396926E 00 0.0000000E 00	0.9396926E 00 0.0000000E 00	0.9396925E 00 0.0000000E 00	0.9396925E 00 0.0000000E 00
2	0.9396926E 00 0.0000000E 00	0.0000000E 00 0.9396926E 00	0.6644629E 00 0.6644628E 00	0.9396926E 00 0.9396925E 00

\*\*\*\*\*

X = 0.1000000E 01

N= 4 NLT= 1 NST= 2 N2T= 0

NC= 1 2 4

NL= 1

NS	DY	DZ
2	0.0000000E 00	0.1000000E 01
3	0.1000000E 01	0.0000000E 00

PT.NO	YB	ZB
1	0.1000000E 01	0.0000000E 00
2	0.1000000E 01	0.0000000E 00
3	0.5000000E 00	0.9216050E 00
4	0.0000000E 00	0.1064178E 01

J	A(J) E(J)	B(J) SGN(J)	C(J) SLO(J)	D(J) SL1(J)
1	0.0000000E 00 0.1000000E 01	0.0000000E 00 1.00000	0.0000000E 00 0.0000000E 00	0.0000000E 00 0.0000000E 00
2	0.1000000E 01 -0.1554019E 01	0.1244857E 00 -1.00000	0.1007751E 01 -0.9396920E 00	-0.1460299E 01 0.1064178E 01

PT.NO	YB(CALC)	ZB(CALC)
1	0.1000000E 01	0.0000000E 00
2	0.9999997E 00	-0.3240224E-06
3	0.4999997E 00	0.9216052E 00
4	0.0000000E 00	0.1064178E 01

J	YY( 3,J,1) YY( 3,J,5)	YY( 3,J,2) YY( 3,J,6)	YY( 3,J,3) YY( 3,J,7)	YY( 3,J,4) YY( 3,J,8)
1	0.1000000E 01 0.0000000E 00	0.1000000E 01 0.0000000E 00	0.1000000E 01 0.0000000E 00	0.1000000E 01 0.0000000E 00
2	0.1000000E 01 0.0000000E 00	0.0000000E 00 0.1064178E 01	0.6886622E 00 0.7716160E 00	0.9999999E 00 0.1064178E 01

\*\*\*\*\*

X = 0.1000000E 02

N= 4 NLT= 1 NST= 2 N2T= 0

NC= 1 2 4

NL= 1

NS	DY	DZ
2	0.000000E 00	0.100000E 01
3	0.100000E 01	0.000000E 00

PT.NO	YB	ZB
1	0.100000E 01	0.000000E 00
2	0.100000E 01	0.327573E 01
3	0.500000E 00	0.419733E 01
4	0.000000E 00	0.433991E 01

J	A(J) E(J)	B(J) SGN(J)	C(J) SL0(J)	D(J) SL1(J)
1	0.000000E 00 0.100000E 01	0.000000E 00 1.00000	0.000000E 00 0.000000E 00	0.000000E 00 0.000000E 00
2	0.100000E 01 -0.1554021E 01	0.1244883E 00 -1.00000	0.1007780E 01 -0.9396911E 00	-0.1460300E 01 0.1064179E 01

PT.NO	YB(CALC)	ZB(CALC)
1	0.100000E 01	0.000000E 00
2	0.9999997E 00	0.3275729E 01
3	0.4999997E 00	0.4197334E 01
4	0.000000E 00	0.4339910E 01

J	YY( 4,J,1) YY( 4,J,5)	YY( 4,J,2) YY( 4,J,6)	YY( 4,J,3) YY( 4,J,7)	YY( 4,J,4) YY( 4,J,8)
1	0.100000E 01 0.000000E 00	0.100000E 01 0.3275730E 01	0.100000E 01 0.1637865E 01	0.100000E 01 0.1637865E 01
2	0.100000E 01 0.3275730E 01	0.000000E 00 0.4339910E 01	0.6886611E 00 0.4047346E 01	0.100000E 01 0.4339909E 01

I J K YXL(I,J,K) YXR(I,J,K) FOR INITIAL SPLINE

1	1	10000000000000000000	0.100000E 21
2	1	1 0.3187567E 00	0.3187567E 00
3	1	1 0.1068660E 00	0.1068660E 00
4	1	1 -0.1074312E 00	-0.1074312E 00
1	1	20000000000000000000	0.100000E 21

2	1	2	0.3187567E 00	0.3187567E 00
3	1	2	0.1068660E 00	0.1068660E 00
4	1	2	-0.1074312E 00	-0.1074312E 00
1	1	3	0.1000000E 21	0.1000000E 21
2	1	3	0.3187572E 00	0.3187572E 00
3	1	3	0.1068666E 00	0.1068666E 00
4	1	3	-0.1074317E 00	-0.1074317E 00
1	1	4	0.1000000E 21	0.1000000E 21
2	1	4	0.3187572E 00	0.3187572E 00
3	1	4	0.1068666E 00	0.1068666E 00
4	1	4	-0.1074317E 00	-0.1074317E 00
1	1	5	0.0000000E 00	0.0000000E 00
2	1	5	0.0000000E 00	0.0000000E 00
3	1	5	0.0000000E 00	0.0000000E 00
4	1	5	0.0000000E 00	0.0000000E 00
1	1	6	0.0000000E 00	0.0000000E 00
2	1	6	-0.1954786E-02	-0.1954786E-02
3	1	6	0.7964261E-02	0.7964261E-02
4	1	6	0.6901021E 00	0.6901021E 00
1	1	7	0.0000000E 00	0.0000000E 00
2	1	7	-0.1070632E-02	-0.1070632E-02
3	1	7	0.4355930E-02	0.4355930E-02
4	1	7	0.3315780E 00	0.3315780E 00
1	1	8	0.0000000E 00	0.0000000E 00
2	1	8	-0.1070632E-02	-0.1070632E-02
3	1	8	0.4355930E-02	0.4355930E-02
4	1	8	0.3315780E 00	0.3315780E 00
1	2	1	0.1000000E 21	0.1000000E 21
2	2	1	0.3187567E 00	0.3187567E 00
3	2	1	0.1068660E 00	0.1068660E 00
4	2	1	-0.1074312E 00	-0.1074312E 00
1	2	2	0.0000000E 00	0.0000000E 00
2	2	2	0.0000000E 00	0.0000000E 00
3	2	2	0.0000000E 00	0.0000000E 00
4	2	2	0.0000000E 00	0.0000000E 00
1	2	3	0.1000000E 21	0.1000000E 21
2	2	3	0.1638253E 00	0.1638253E 00
3	2	3	0.2244857E-01	0.2244857E-01
4	2	3	-0.2192813E-01	-0.2192813E-01
1	2	4	0.1000000E 21	0.1000000E 21
2	2	4	0.3187567E 00	0.3187567E 00
3	2	4	0.1068656E 00	0.1068656E 00
4	2	4	-0.1074307E 00	-0.1074307E 00

1	2	5UUUUUUUUUUUUUUUUUU	0.0000000E 00
2	2	5 -0.1954786E-02	-0.1954786E-02
3	2	5 0.7964261E-02	0.7964261E-02
4	2	5 0.6901021E 00	0.6901021E 00

1	2	6UUUUUUUUUUUUUUUUUU	0.1000000E 21
2	2	6 0.4625672E 00	0.4625672E 00
3	2	6 0.3173745E 00	0.3173745E 00
4	2	6 0.4117222E 00	0.4117222E 00

1	2	7UUUUUUUUUUUUUUUUUU	0.1000000E 21
2	2	7 0.3236457E 00	0.3236457E 00
3	2	7 0.3088703E 00	0.3088703E 00
4	2	7 0.4187793E 00	0.4187793E 00

1	2	8UUUUUUUUUUUUUUUUUU	0.1000000E 21
2	2	8 0.4625650E 00	0.4625650E 00
3	2	8 0.3173720E 00	0.3173720E 00
4	2	8 0.4117249E 00	0.4117249E 00

J= 1 K= 1 MLT= 1 MST= 0 M2T= 0 MIT= 1

MI= 1

ML= 3

I YXL(I, 1, 1) YXR(I, 1, 1)

1	UUUUUUUUUUUUUUUUUU	0.1000000E 21
2	0.3639706E 00	0.3639706E 00
3	0.0000000E 00	0.0000000E 00
4	0.0000000E 00	-0.1074312E 00

J= 1 K= 2 MLT= 1 MST= 0 M2T= 0 MIT= 1

MI= 1

ML= 3

I YXL(I, 1, 2) YXR(I, 1, 2)

1	UUUUUUUUUUUUUUUUUU	0.1000000E 21
2	0.3639706E 00	0.3639706E 00
3	0.0000000E 00	0.0000000E 00
4	0.0000000E 00	-0.1074312E 00

J= 1 K= 3 MLT= 1 MST= 0 M2T= 0 MIT= 1

MI= 1

ML= 3

I YXL(I, 1, 3) YXR(I, 1, 3)

1	UUUUUUUUUUUUUUUUUU	0.1000000E 21
2	0.3639714E 00	0.3639714E 00
3	0.0000000E 00	0.0000000E 00
4	0.0000000E 00	-0.1074317E 00

J= 1 K= 4 MLT= 1 MST= 0 M2T= 0 MIT= 1

MI= 1

ML= 3

I YXL(I, 1, 4) YXR(I, 1, 4)

1	UUUUUUUUUUUUUUUUUU	0.1000000E 21
2	0.3639714E 00	0.3639714E 00
3	0.0000000E 00	0.0000000E 00
4	0.0000000E 00	-0.1074317E 00

J= 1 K= 6 MLT= 3 MST= 0 M2T= 0 MIT= 0

ML= 1 2 3

I YXL(I, 1, 6) YXR(I, 1, 6)

1	UUUUUUUUUUUUUUUUUU	0.0000000E 00
2	0.0000000E 00	0.0000000E 00
3	0.0000000E 00	0.3639700E 00
4	0.3639700E 00	0.6901021E 00

J= 1 K= 7 MLT= 3 MST= 0 M2T= 0 MIT= 0

ML= 1 2 3

I YXL(I, 1, 7) YXR(I, 1, 7)

1	UUUUUUUUUUUUUUUUUU	0.0000000E 00
2	0.0000000E 00	0.0000000E 00
3	0.0000000E 00	0.1819850E 00
4	0.1819850E 00	0.3315780E 00

J= 1 K= 8 MLT= 3 MST= 0 M2T= 0 MIT= 0

MLP 1 2 3

I YXL(I, 1, 8) YXR(I, 1, 8)

1	UUUUUUUUUUUUUUUUUU	0.0000000E 00
2	0.0000000E 00	0.0000000E 00
3	0.0000000E 00	0.1819850E 00
4	0.1819850E 00	0.3315780E 00

J= 2 K= 1 MLT= 1 MST= 0 M2T= 0 MIT= 1

MLP 1

MLD 3

I YXL(I, 2, 1) YXR(I, 2, 1)

1	UUUUUUUUUUUUUUUUUU	0.1000000E 21
2	0.3639706E 00	0.3639706E 00
3	0.0000000E 00	0.0000000E 00
4	0.0000000E 00	-0.1074312E 00

J= 2 K= 3 MLT= 1 MST= 0 M2T= 0 MIT= 1

MLP 1

MLD 3

I YXL(I, 2, 3) YXR(I, 2, 3)

1	UUUUUUUUUUUUUUUUUU	0.1000000E 21
2	0.1726953E 00	0.1726953E 00
3	-0.1192093E-06	-0.1192093E-06
4	-0.1192093E-06	-0.2192813E-01

J= 2 K= 4 MLT= 1 MST= 0 M2T= 0 MIT= 1

MLP 1

MLD 3

I YXL(I, 2, 4) YXR(I, 2, 4)

1	UUUUUUUUUUUUUUUUUU	0.1000000E 21
2	0.3639705E 00	0.3639705E 00



3 0.1324547E-07 0.1324548E-07  
 4 0.1324548E-07 -0.1074307E 00

J= 2 K= 5 MLT= 3 MST= 0 M2T= 0 MIT= 0

ML= 1 2 3

I YXL(I, 2, 5) YXR(I, 2, 5)

1	UUUUUUUUUUUUUUUUUU	0.0000000E 00
2	0.0000000E 00	0.0000000E 00
3	0.0000000E 00	0.3639700E 00
4	0.3639700E 00	0.6901021E 00

J= 2 K= 6 MLT= 2 MST= 0 M2T= 0 MIT= 1

MI= 1

ML= 2 3

I YXL(I, 2, 6) YXR(I, 2, 6)

1	UUUUUUUUUUUUUUUUUU	0.1000000E 21
2	0.3639724E 00	0.3639724E 00
3	0.3639724E 00	0.3639701E 00
4	0.3639701E 00	0.4117222E 00

J= 2 K= 7 MLT= 1 MST= 0 M2T= 0 MIT= 1

MI= 1

ML= 3

I YXL(I, 2, 7) YXR(I, 2, 7)

1	UUUUUUUUUUUUUUUUUU	0.1000000E 21
2	0.3046873E 00	0.3046873E 00
3	0.3639699E 00	0.3639700E 00
4	0.3639700E 00	0.4187793E 00

J= 2 K= 8 MLT= 2 MST= 0 M2T= 0 MIT= 1

MI= 1

ML= 2 3

I YXL(1, 2, 8) YXR(1, 2, 8)

10000000000000000000 0.1000000E 21  
 2 0.3639699E 00 0.3639700E 00  
 3 0.3639700E 00 0.3639701E 00  
 4 0.3639701E 00 0.4117249E 00

I J A(1) A(2) A(3)  
 A(4) A(5)

2 2 -0.1132472E 01 -0.9536743E-06 -0.1132476E 01  
 -0.2086741E-05 0.3235974E-06

3 2 -0.1000010E 01 -0.8583069E-05 -0.8830320E 00  
 0.9797735E-05 0.1088716E-04

4 1 0.0000000E 00 0.0000000E 00 0.0000000E 00  
 -0.1000000E 01 0.0000000E 00

4 2 0.1179822E 00 0.2646446E-04 0.1041781E 00  
 -0.1144409E-03 -0.6825432E 00

X FP	PHI FXX	F FPP	FX FXP
0.3000000E 00 -0.4168083E-06	0.0000000E 00 -0.1721457E 01	0.7141427E 00 0.6284285E-01	0.4789243E 00 -0.1111489E-05
0.3000000E 00 0.9015270E-02	0.5235987E 00 -0.1754072E 01	0.7185342E 00 -0.2382276E-01	0.4492615E 00 -0.6265515E-01
0.3000000E 00 -0.9016611E-02	0.1047197E 01 -0.1754071E 01	0.7185337E 00 -0.2382181E-01	0.4492621E 00 0.6265879E-01
0.3000000E 00 -0.3241843E-06	0.1570796E 01 -0.1721455E 01	0.7141420E 00 0.6284434E-01	0.4789269E 00 0.3704963E-05
0.1000000E 01 0.1152045E-05	0.0000000E 00 -0.1169780E 00	0.9999990E 00 0.1169729E 00	-0.2955627E-06 0.3213953E 00
0.1000000E 01 0.5295887E-01	0.5235987E 00 -0.9172714E-01	0.1014950E 01 0.6944185E-01	0.1655378E 00 0.3039948E 00
0.1000000E 01 0.5813339E-01	0.1047197E 01 -0.3356271E-01	0.1046981E 01 -0.5744129E-01	0.3051024E 00 0.2100322E 00
0.1000000E 01 -0.3316336E-06	0.1570796E 01 0.4265173E-06	0.1064177E 01 -0.1409823E 00	0.3639702E 00 0.9466912E-06
0.2000000E 01 0.3214021E 00	0.0000000E 00 -0.1409802E 00	0.9396916E 00 0.2198226E 00	-0.1244882E 00 0.3214030E 00

0.2000000E 01	0.5235987E 00	0.1133538E 01	0.6936222E-01
0.4050109E 00	-0.1057327E 00	0.5557603E-01	0.4050160E 00
0.2000000E 01	0.1047197E 01	0.1335164E 01	0.2709887E 00
0.3244193E 00	-0.3524287E-01	-0.4042364E 00	0.3244169E 00
0.2000000E 01	0.1570796E 01	0.1428148E 01	0.3639703E 00
0.3481045E-05	0.1960301E-06	-0.7423788E 00	0.3281959E-06
0.5000000E 01	0.0000000E 00	0.4032016E-04	0.3095926E 02
-0.4086256E 07	0.7555792E 14	0.1316286E 25	-0.9972746E 19
0.5000000E 01	0.5235987E 00	0.6621871E 00	0.2936798E 06
-0.1827027E 07	0.2014248E 18	0.7795724E 19	-0.1253096E 19
0.5000000E 01	0.1047197E 01	0.1951118E 01	0.1127432E 00
0.2170684E 01	-0.9872645E-01	-0.4567570E 01	0.1183946E 01
0.5000000E 01	0.1570796E 01	0.2520061E 01	0.3639699E 00
0.1118774E-04	-0.6192974E-05	-0.4238317E 01	0.4148762E-05

OBJECT CODE= 47112 BYTES,ARRAY AREA= 24692 BYTES

## REFERENCES

1. Bartlett, D. A.: Computer Utilization for Aircraft Contour Determination. Society of Automotive Engineers, 700202, National Business Meeting, Wichita, Kansas, March 18-20, 1970.
2. Coons, S. A.: Surfaces for Computer-Aided Design of Space Forms. AD663504 MAC-TR-41, MIT, June, 1967.
3. DeJarnette, F.R.: Calculation of Inviscid Surface Streamlines and Heat Transfer on Shuttle Type Configurations. NASA CR-111921, Aug., 1971.
4. Rakich, J.V.; and Kutler, P.: Comparison of Characteristics and Shock Capturing Methods with Application to the Space Shuttle Vehicle. AIAA Paper No. 72-191, Jan. 1972.
5. DeJarnette, F. R.: A Least-Square Curve-Fitting Technique Using General Conic Sections. Paper written while attending NASA-ASEE Summer Faculty Program at Langley Research Center, Aug., 1972.
6. Bezier, P.: Numerical Control Mathematics and Applications. John Wiley and Sons, 1972.
7. Mason, T. E., and Hazard, C. T.: Brief Analytic Geometry. Ginn and Co., 1947.
8. Scheid, F.: Theory and Problems of Numerical Analysis. Schaum's Outline Series, McGraw-Hill Book Co., 1968.
9. Ahlberg, J. H., Nilson, E. N., and Walsh, J. L.: The Theory of Splines and Their Applications. Academic Press, 1967.

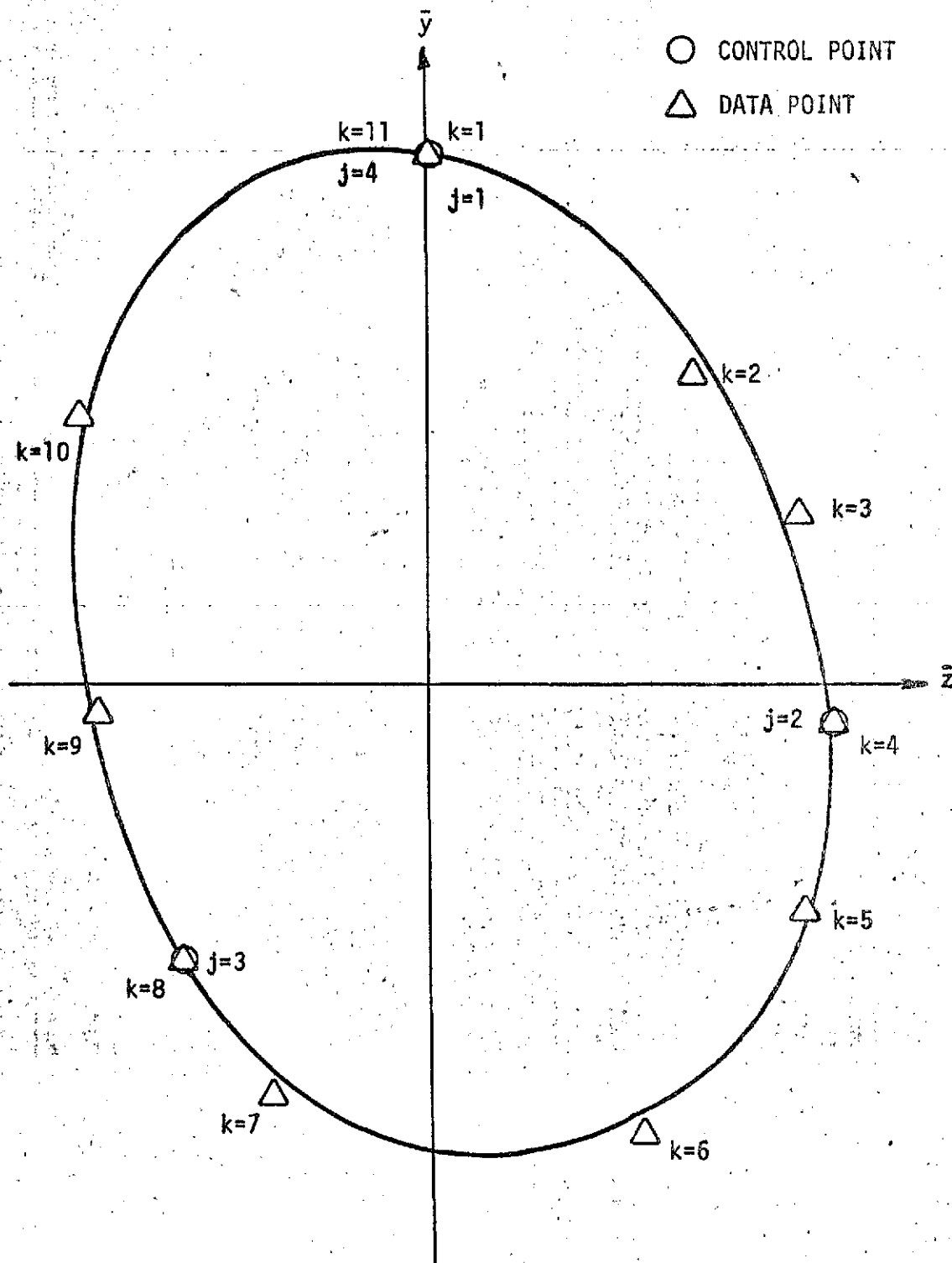


Figure 1. Control points and data points in a cross section.

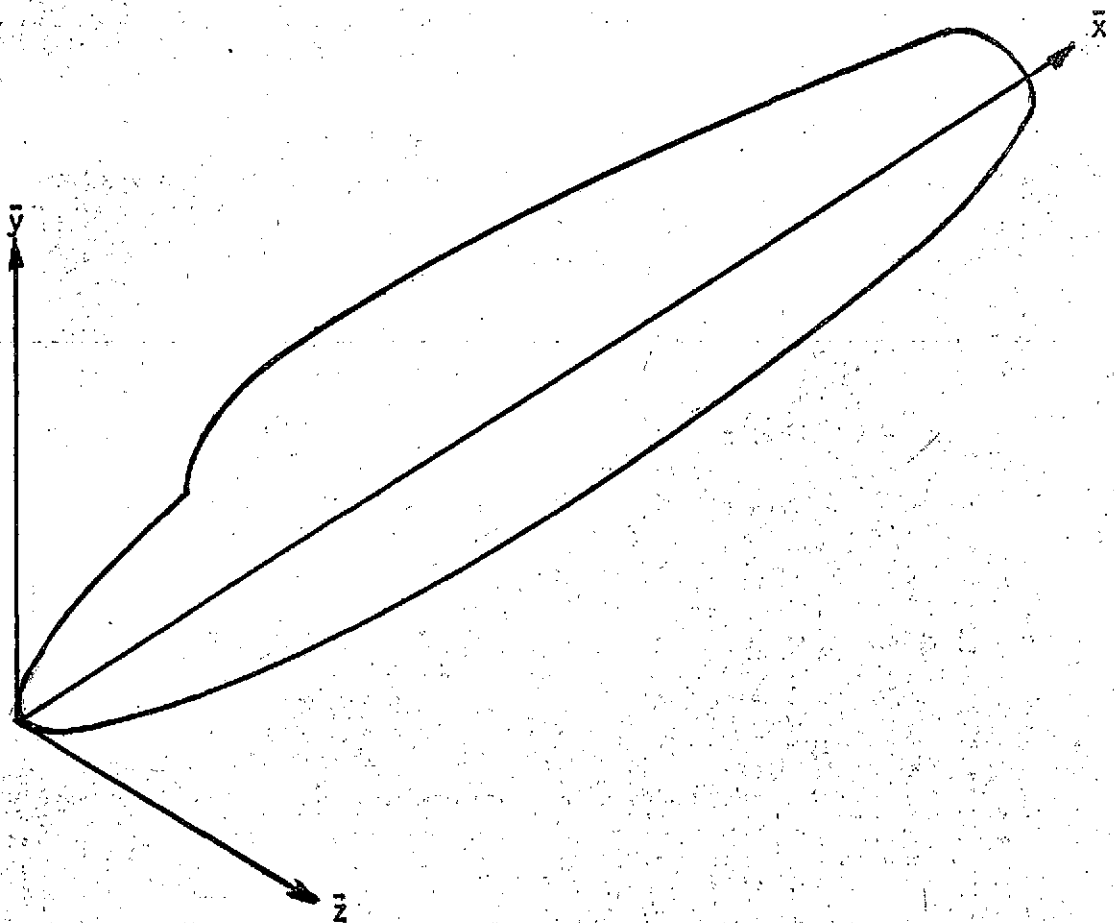


Figure 2. Cartesian coordinate system.

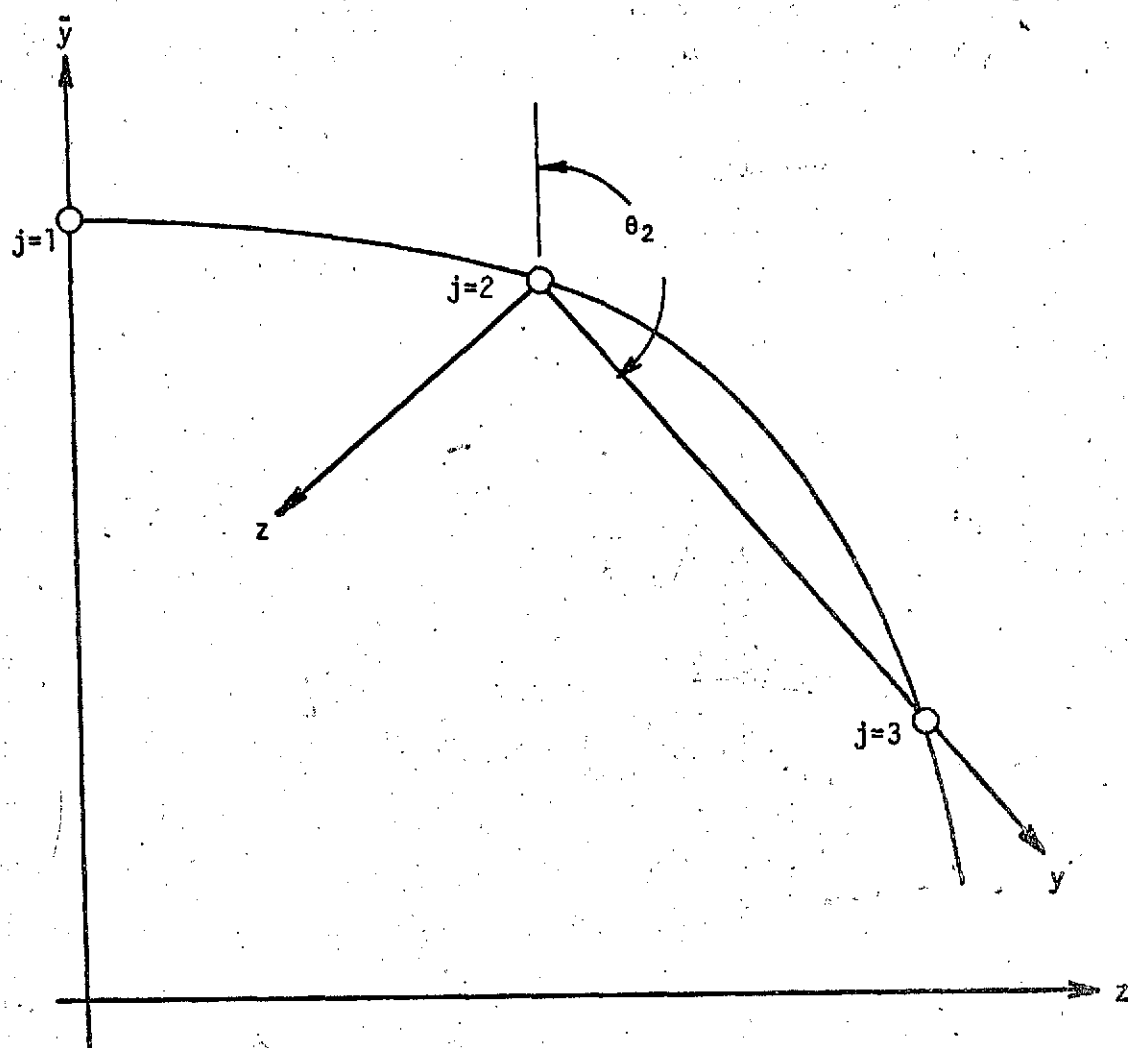


Figure 3. Local c-ordinate system, illustrated for segment  $j=2$ .

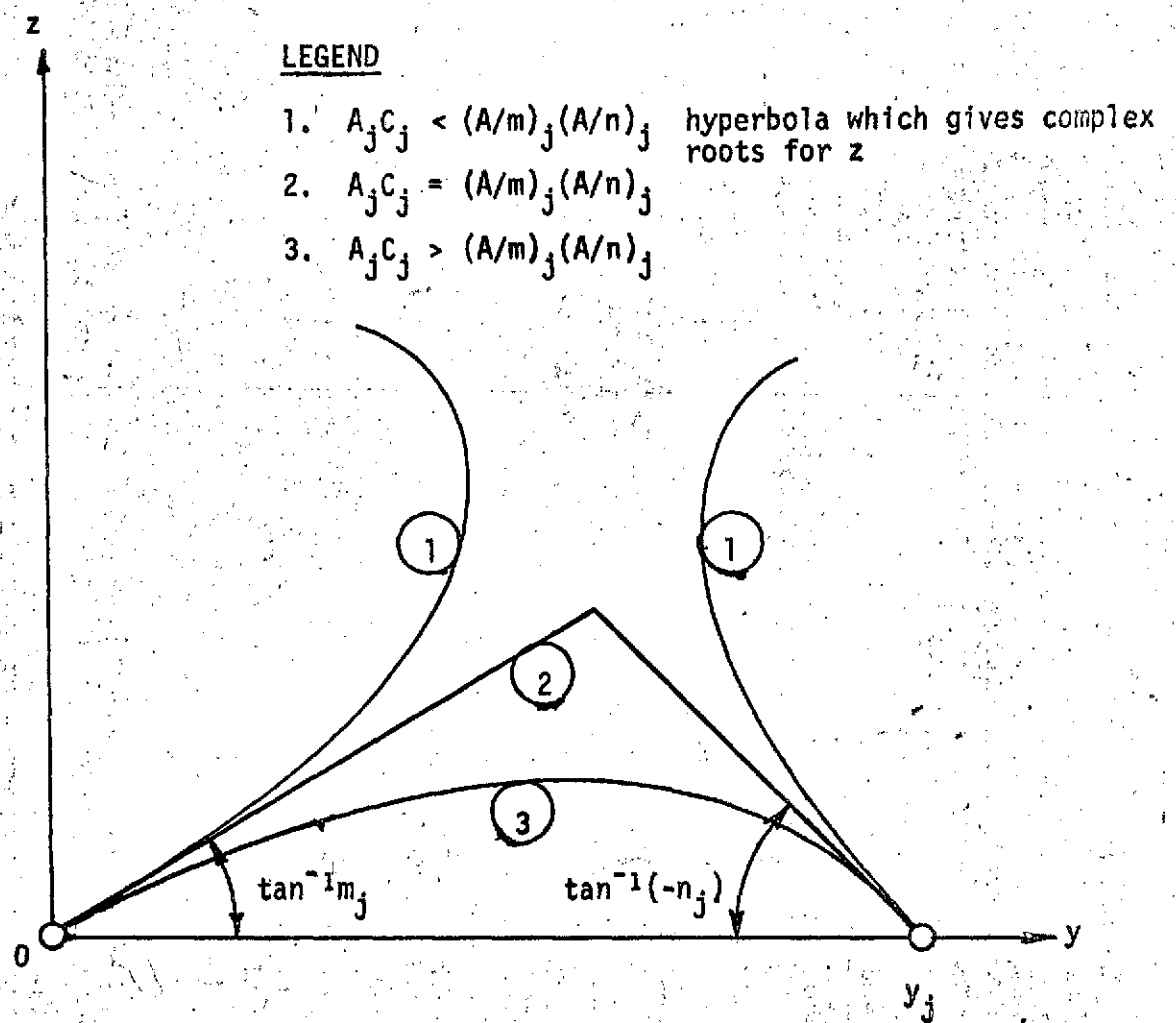
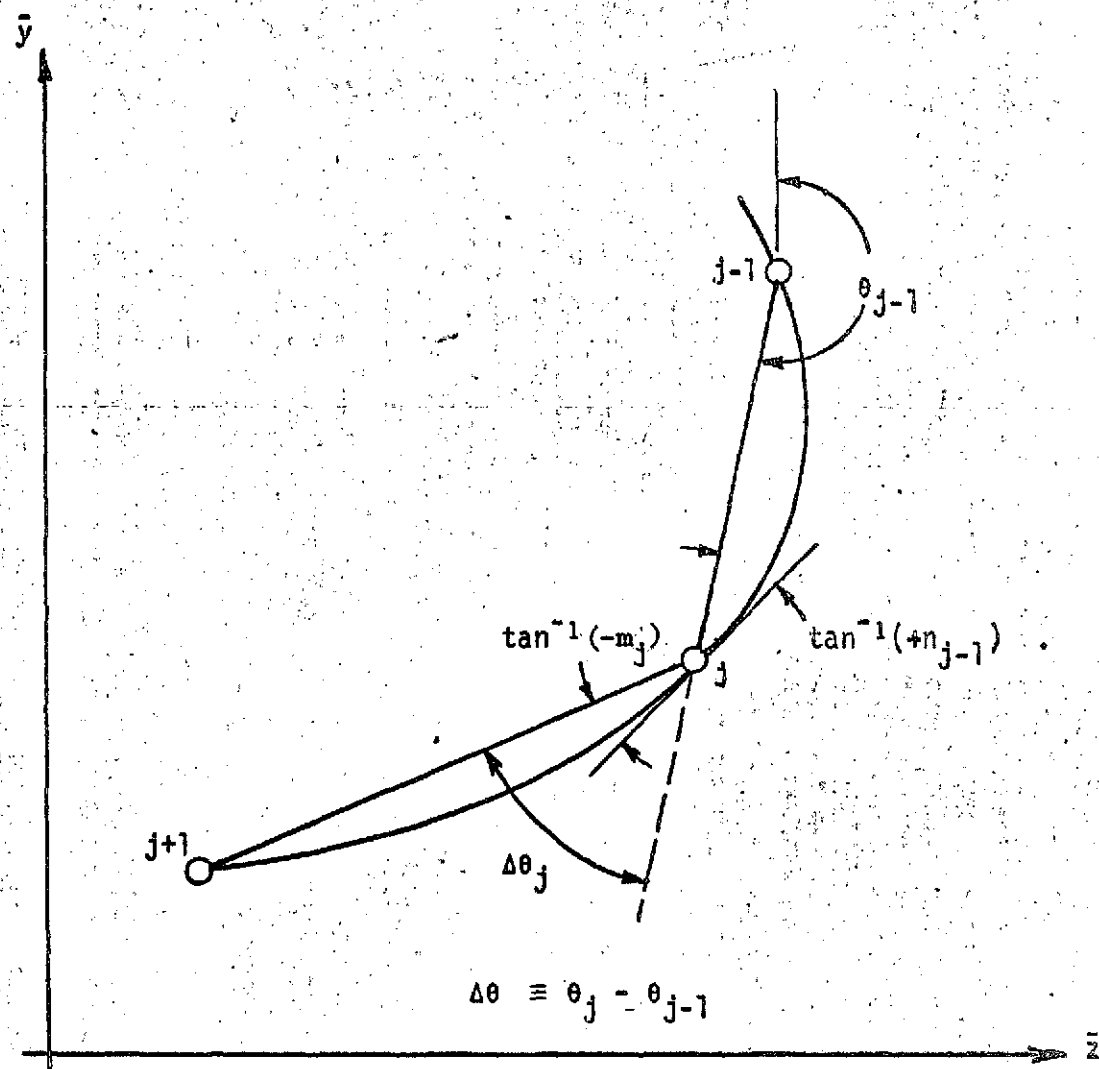


Figure 4. Effect of  $A_j C_j$  on conic section.





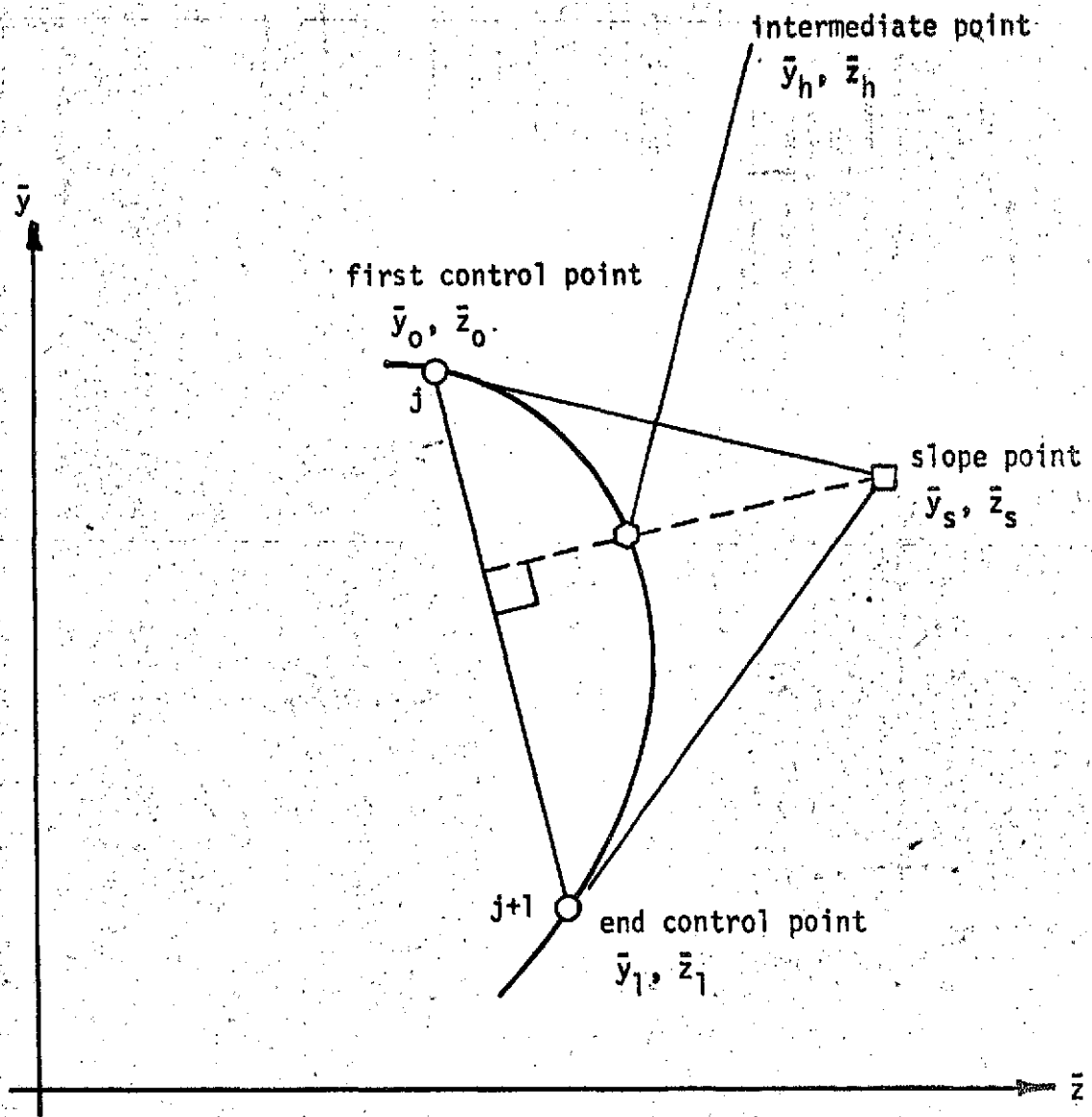


Figure 6. Four points used to define a segment of a conic section.

- CONTROL POINT,  $(\bar{y}_0, \bar{z}_0)$  and  $(\bar{y}_1, \bar{z}_1)$
- ◇ INTERMEDIATE POINT,  $(\bar{y}_h, \bar{z}_h)$
- SLOPE POINT,  $(\bar{y}_s, \bar{z}_s)$

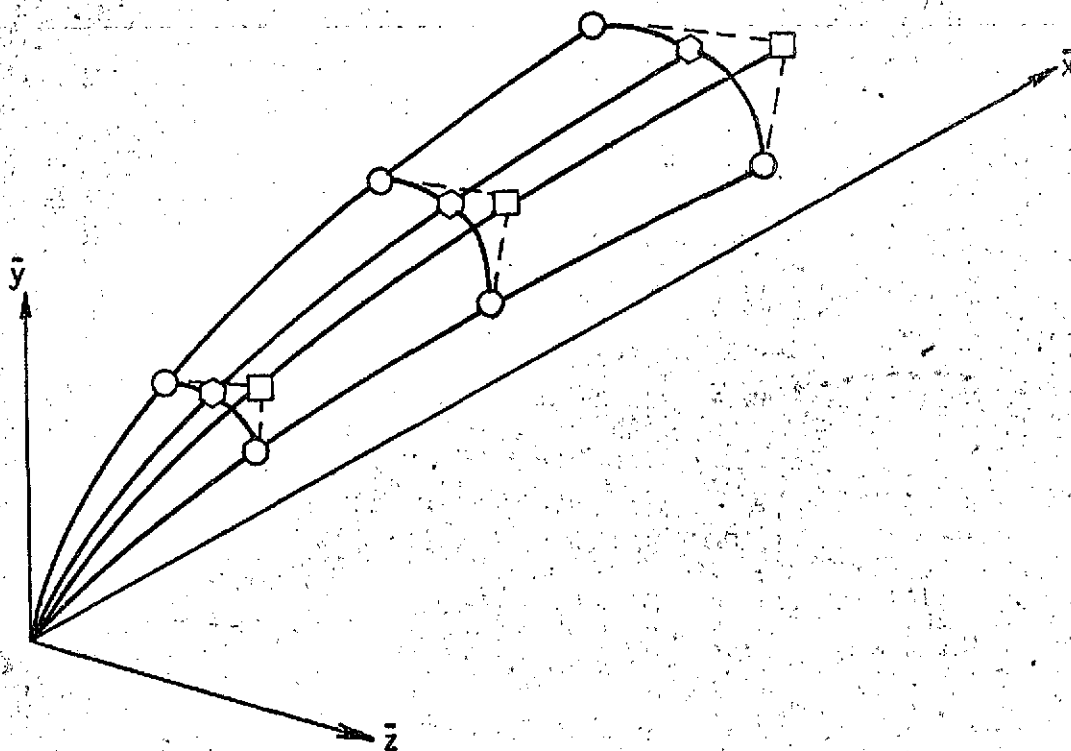


Figure 7. Longitudinal curves through the four points used to define a segment of a conic section.

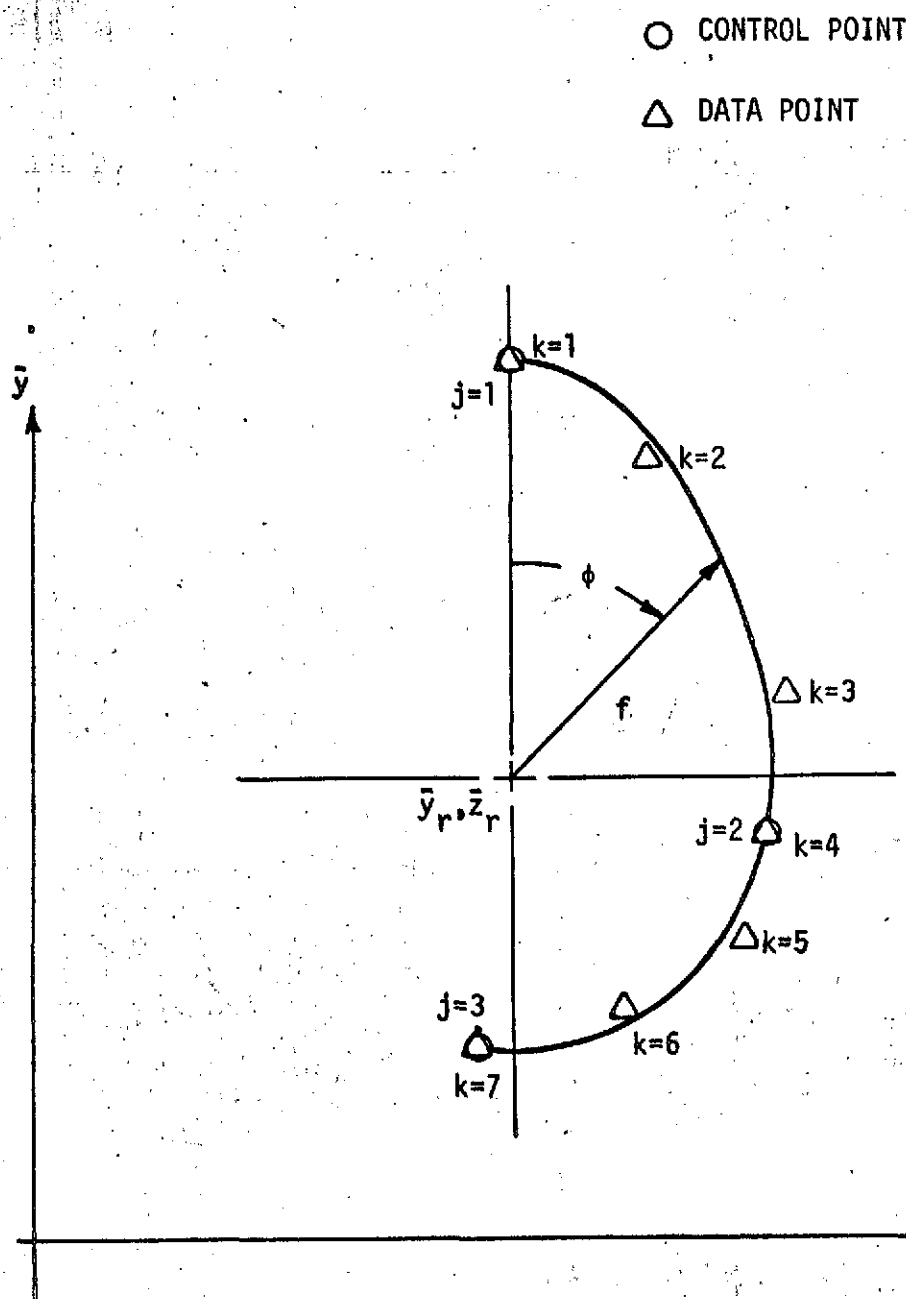
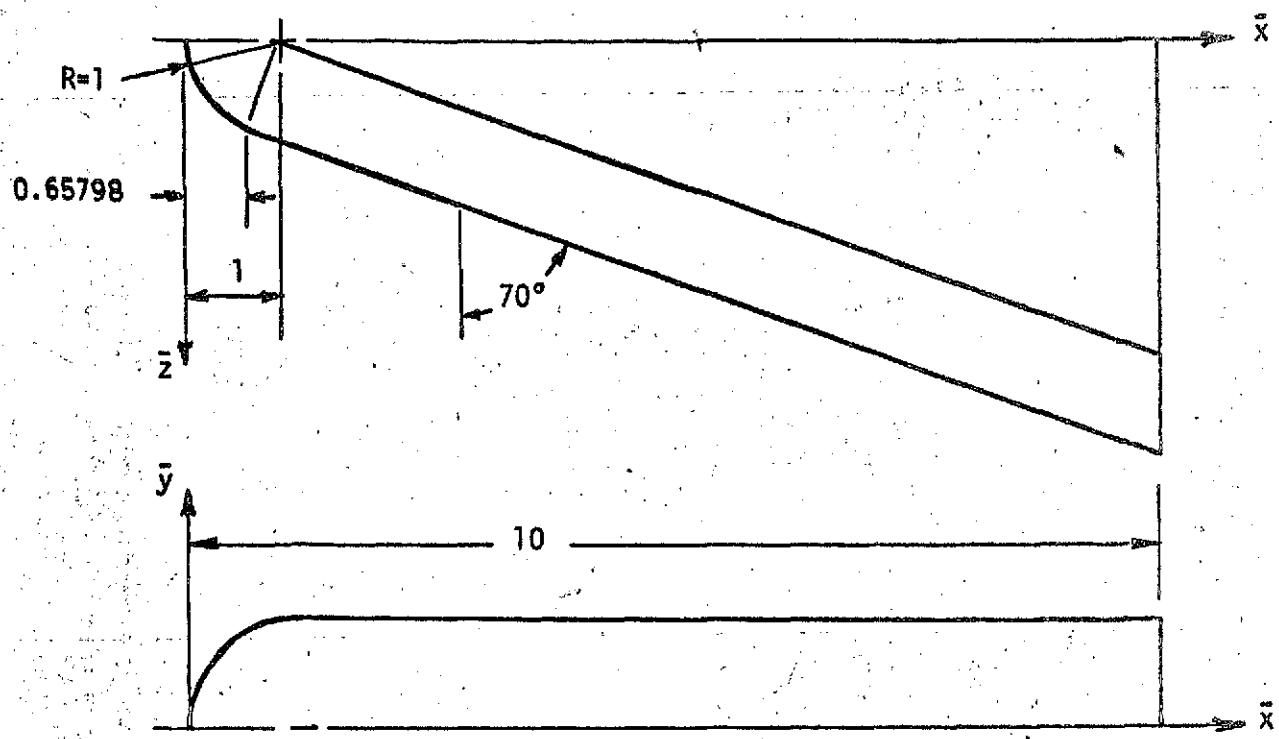


Figure 8. Polar coordinates.



○ CONTROL POINT  
 △ DATA POINT

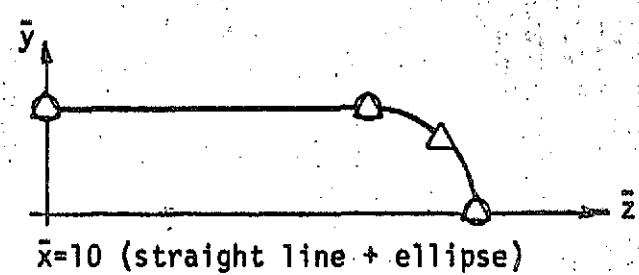
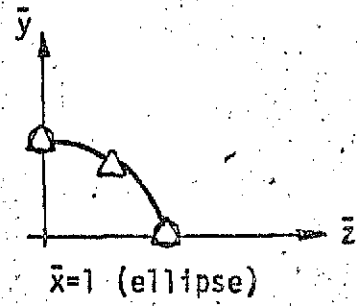
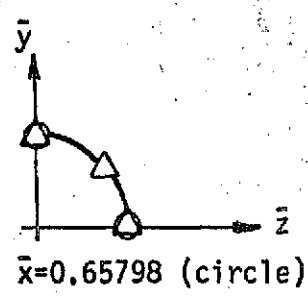


Figure 9. Geometry of 70° delta wing.